

Finite Element Computations of Transonic Viscous Flows with the Streamline Upwind Petrov-Galerkin (SUPG) Formulation

Constantin Bucur

Department of Mechanical Engineering
McGill University
Montréal, Québec, Canada

February 2006

A thesis submitted to McGill University in partial fulfillment of the requirements of
the degree of Master of Engineering (THESIS)

© Copyright 2006, Constantin Bucur



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-24944-4

Our file Notre référence

ISBN: 978-0-494-24944-4

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

Computations of transonic viscous flows are very challenging. The major difficulty comes from the discontinuity in the solution across a shock wave, causing undesired oscillations in the solution. In this work we focus on minimizing the oscillations by the use of a limiter to control the amount of diffusivity. This limiter provides the right amount of viscosity to capture a sharp shock and an accurate solution in high gradient regions. The limiter employs changes in pressure and entropy and has been implemented into the Streamline Upwind Finite Element Method. A mesh adaptation strategy has been employed to further enhance the accuracy of the solution. Results of simulations over RAE 2822 airfoil and ONERA M6 wing indicate significant improvements to the solution with this implementation.

Résumé

Les calculs des écoulements visqueux transsoniques constituent un grand défi. La difficulté principale vient de la discontinuité dans la solution à travers une onde de choc, causant des oscillations non désirées dans la solution. Dans ce travail nous nous concentrons à réduire au minimum les oscillations par l'utilisation d'un limiteur pour contrôler la quantité de diffusivité. Ce limiteur fournit la bonne quantité de viscosité pour capturer un choc propre et une solution précise dans des régions de gradient élevées. Le limiteur est fonction des changements de pression et d'entropie et a été implémenté dans une méthode élément fini décentrée de type SUPG. Une stratégie d'adaptation de maillage a été utilisée pour augmenter davantage l'exactitude de la solution. Des résultats de simulations sur l'aile RAE 2822 et l'aile d'ONERA M6 montrent des améliorations significatives de la solution avec l'approche proposée.

Acknowledgement

I would first like to thank Professor Siva Nadarajah for his support, guidance and patience shown during the course of this research and for providing an excellent role model to pursue in the years to come. I am also grateful to Professor W.G. Habashi for his many valuable suggestions.

I am greatly indebted to my CFD Lab colleague Dr. Lakhdar Remaki for supplying the mesh adaptation code, as well as many valuable discussions and insightful interventions throughout the program. I also like to thank Dr. Kasidit Leoviriyakit for his help with the presentation preparation.

This work was partially supported by a National Science and Engineering Research Council of Canada (NSERC) Discovery Grant, and its role is duly acknowledged.

To my family, my eternal thanks for their continuous encouragement and support during my studies.

Last but not least, I would like to specially thank Iuliana for her support, patience, and words of encouragement during my studies.

Table of Contents

Abstract.....	II
Résumé.....	III
Acknowledgement.....	IV
Table of Contents.....	V
List of Figures.....	VII
List of Tables.....	IX
Nomenclature.....	X
1. Introduction.....	1
1.1. Dissertation directions.....	6
2. Governing Equations.....	7
2.1. Navier Stokes equations in differential form.....	7
2.1.1. Continuity equation.....	8
2.1.2. Momentum equations.....	8
2.1.3. Energy equation.....	10
2.2. Navier Stokes equations in conservation form.....	10
2.3. Reference variables.....	12
2.4. Spalart-Allmaras turbulence model.....	13
2.5. Boundary conditions (BC).....	14
2.6. Finite Element Method (FEM) formulation.....	15
2.7. Artificial viscosity (AV).....	17
2.8. FENSAP – solving strategy.....	18
3. Methodology.....	20
3.1. Streamline Upwind (SU) and SU Petrov-Galerkin (SUPG).....	20
3.1.1. The stabilization of one-dimensional convective dominated flow.....	21
3.1.2. Upwind and Streamline Upwind stabilisation.....	22
3.1.3. Streamline Upwind – Petrov Galerkin stabilisation.....	24
3.2. Shock wave detector definition.....	28
3.2.1. Shock detector for 2D inviscid flow.....	28
3.2.2. Shock Sensor implemented in FENSAP.....	30
3.2.3. Entropy based shock detection.....	31
3.3. Streamline and Cross Wind directions definition.....	33
3.4. Artificial Viscosity Switching Scheme.....	34
3.5. Mesh generation.....	37
3.6. Mesh adaptation.....	38
4. Numerical results - RAE 2822 airfoil test case.....	46
4.1. Geometry.....	46
4.2. Flow parameters.....	46
4.3. Computation mesh.....	47
4.4. Discussion of results.....	48
4.4.1. SU scheme without shock detector.....	49
4.4.1.1. Fully first order AV scheme.....	49
4.4.1.2. Fully second order AV scheme.....	51
4.4.2. SU scheme with shock detector.....	53
4.5. Comparison between SU and SU with shock detector.....	54
5. Numerical Results - ONERA M6 wing test case.....	69
5.1. Wing's Geometry.....	69
5.2. Flow parameters.....	70
5.3. Computation meshes.....	70
5.4. Discussion of results.....	71
5.4.1. Initial solution.....	73

5.4.2. Restart solution before mesh adaptation	74
5.4.2.1. Solution obtained using a shock sensor based on pressure.....	75
5.4.2.2. Solution obtained using a shock sensor based on entropy.....	75
5.4.3. Solution after mesh adaptation.....	75
6. Conclusions	102
6.1. Future work	103
Bibliography	105
7. Appendix A – RAE 2822 measured coordinates, [16].....	109
8. Appendix B – ONERA M6 wing data, [5].....	111

List of Figures

Figure 1-1: Prandtl-Glauert condensation cloud on a transonic F/A-18 Hornet plane (media release, USS Constellation, July 7, 1999 [990707-N-6483G-001])	2
Figure 3-1: Comparison between linear Galerkin and Petrov-Galerkin weights	40
Figure 3-2: Arbitrary 2D mesh used to illustrate the discontinuity detector definition for a finite volume scheme	40
Figure 3-3: Arbitrary unstructured 2D mesh used to illustrate shock detector, local and global derivative definitions	41
Figure 3-4: Global and local derivatives for 1D flow.....	41
Figure 3-5: Streamline and cross wind directions at mesh point A	42
Figure 3-6: Artificial Viscosity ratios as defined in FENSAP Graphical User Interface (from NTI – fensapiceGUI executable)	42
Figure 3-7: Switching procedure	43
Figure 3-8: (a) Shock sensor contours for a 2D transonic viscous flow around RAE 2822 airfoil. (b) Shock sensor variation through the shock wave	44
Figure 3-9: RAE 2822 transonic viscous flow: (a) Sensor contours for threshold $S=0.99$ –first order dominated AV coefficient (b) Sensor contours for threshold $S=0.9$ –second order dominated AV coefficient.....	45
Figure 4-1: (a) RAE 2822 Hybrid Mesh (tetras and prisms). (b) Airfoil detail.....	56
Figure 4-2: (a) RAE 2822 Leading edge detail. (b) RAE 2822 Trailing edge detail.	57
Figure 4-3: RAE 2822 transonic viscous solution using tetra mesh and SU fully first order AV scheme. (a) Mach contours. (b) Shock wave detail.	58
Figure 4-4: RAE 2822 transonic viscous solution using tetra mesh and SU fully first order AV scheme. (a) Wall Coefficient of Pressure distribution – experimental and present solution data. (b) Mach number variation through the shock wave.....	59
Figure 4-5: Overall Residuals for fully second order AV case: (a) Simulation A. (b) Simulation B.	60
Figure 4-6: C_l and C_d convergence for fully second order case: (a) Coefficient of lift convergence for the restart solutions A and B. (b) Coefficient of drag convergence for the restart solution A and B.	61
Figure 4-7: RAE 2822 transonic viscous solution using tetra mesh and SU fully second order AV scheme. (a) Mach contours. (b) Shock wave detail.	62
Figure 4-8: RAE 2822 transonic viscous solution using tetra mesh and SU fully second order AV scheme. (a) Wall Coefficient of Pressure distribution – experimental and present solution data. (b) Mach number variation through the shock wave.....	63
Figure 4-9: Histograms - RAE 2822 transonic viscous solution using tetra mesh and SU fully second order AV scheme. (a) Overall residual. (b) Spalart-Allmaras residual.....	64
Figure 4-10: (a) Mach contours of the RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme. (b) Zoom in the shock region of (a).	65
Figure 4-11: RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme. (a) C_p distribution at the wall – present solution against experimental data. (b) Mach number variation through the shock wave.	66
Figure 4-12: RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme - Histograms: (a) Coefficient of lift C_l . (b) Coefficient of drag C_d	67
Figure 4-13: RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme - Histograms for Overall residual and Spalart-Allmaras residual.....	68
Figure 4-14: C_p comparison for SU fully first order, SU fully second order and SU + shock sensor.	68
Figure 5-1: Initial mesh (i.e. before adaptation) (a) Upper surface view. (b) Wing root detail.	79

Figure 5-2: (a) Top view of the mesh obtained after 1 st adaptation. (b) Top view of the mesh obtained after 2 nd adaptation.	80
Figure 5-3: (a) Mesh obtained after 1 st adaptation – view at 80% wing semi span. (b) Mesh obtained after 2 nd adaptation – view at 80% wing semi span.	81
Figure 5-4: (a) Top view of the mesh obtained after 3 rd adaptation. (b) Mesh obtained after 3 rd adaptation – view at the wing root.	82
Figure 5-5: (a) Mesh obtained after 3 rd adaptation – view at 80% wing semi span. (b) Mesh obtained after 3 rd adaptation – view at 95% wing semi span.	83
Figure 5-6: Mesh adaptation procedure	84
Figure 5-7: Initial solution obtained for F=1 (a) C_p distribution on the wing. (b) Mach number contours at 80% of wing semi span.	85
Figure 5-8: Initial solution obtained for F=1 (a) C_l and C_d coefficients. (b) Wall pressure coefficient distribution at 80% wing semi span.	86
Figure 5-9: Solution obtained for F=4, isotropic grid and pressure shock sensor. (a) C_p distribution on the wing – top view. (b) Mach contours at 80% wing semi span.	87
Figure 5-10: Solution obtained for F=4, isotropic grid and pressure shock sensor. (a) C_l and C_d histograms. (b) C_p distribution comparison.	88
Figure 5-11: Solution obtained for F=4, isotropic grid and entropy shock sensor. (a) C_p contours on the wing – top view. (b) Mach contours at 80% of the wing semi span.	89
Figure 5-12: (a) Solution obtained for F=4, isotropic grid and entropy shock sensor - C_p distribution comparison. (b) C_p comparison between the two shock sensor types.	90
Figure 5-13: 1 st adaptation cycle solution: (a) C_p contours on the wing – top view. (b) Wall C_p comparison at 80% of wing semi span.	91
Figure 5-14: 2 nd adaptation cycle solution: (a) C_p contours on the wing – top view. (b) Wall C_p comparison at 80% of wing semi span.	92
Figure 5-15: 3 rd adaptation cycle solution: (a) Isometric view of the wing; C_p distribution on the wing, and Mach contours in the symmetry plane. (b) Top view of the wing with C_p contours.	93
Figure 5-16: 3 rd adaptation cycle solution; C_p distribution at the wall – comparison: (a) 20% of the wing semi span. (b) 44% of the wing semi span.	94
Figure 5-17: 3 rd adaptation cycle solution; C_p distribution at the wall – comparison: (a) 65% of the wing semi span. (b) 80% of the wing semi span.	95
Figure 5-18: 3 rd adaptation cycle solution; C_p distribution at the wall – comparison: (a) 90% of the wing semi span. (b) 95% of the wing semi span.	96
Figure 5-19: 3 rd adaptation cycle solution; (a) C_p distribution at the wall, 99% of the wing semi span – comparison. (b) Mach contours at 99% of the wing semi span.	97
Figure 5-20: 3 rd adaptation cycle solution; (a) Mach contours at 20% of the wing semi span. (b) Mach contours at 44% of the wing semi span.	98
Figure 5-21: 3 rd adaptation cycle solution; (a) Mach contours at 65% of the wing semi span. (b) Mach contours at 80% of the wing semi span.	99
Figure 5-22: 3 rd adaptation cycle solution; (a) Mach contours at 90% of the wing semi span. (b) Mach contours at 95% of the wing semi span.	100
Figure 5-23: 3 rd adaptation cycle solution; (a) C_l and C_d histograms. (b) Residuals histograms.	101
Figure 7-1: RAE 2822 transonic airfoil – model cross section, [16]	110
Figure 8-1: Design values for root section coordinates of the symmetrical profile	111
Figure 8-2: ONERA M6 semi-span wing	112
Figure 8-3: Additional remarks for ONERA M6 wing	112

List of Tables

Table 4-1: RAE 2822 reference dimensions	46
Table 4-2: Reference conditions for RAE 2822 transonic simulations.....	47
Table 4-3: AV coefficients corresponding to each parameter cycle of simulations A and B.....	50
Table 4-4: Aerodynamic coefficients for the iterations 2000, 2500 and 3000	52
Table 4-5: Relative error for aerodynamic coefficients.....	52
Table 4-6: C_l and C_d from experiment and present solution (hybrid mesh and SU + shock detector)	54
Table 4-7: C_l and C_d for other CFD and for present solution (hybrid mesh and SU + shock detector)	54
Table 4-8: C_l and C_d comparison.....	54
Table 5-1: ONERA M6 wing reference dimensions	69
Table 5-2: ONERA M6 wing flight conditions	70
Table 5-3: Meshes properties.....	71
Table 5-4: Comparison between present mesh and other CFD meshes.....	71
Table 5-5: AV coefficients corresponding to each parametric cycle - initial solution.	74
Table 5-6: ONERA M6 – aerodynamic coefficients for intermediate solution.....	75
Table 5-7: Aerodynamic coefficients values for each solution	77
Table 5-8: Relative error for aerodynamic coefficients.....	77
Table 5-9: Aerodynamic coefficients for 3 rd adaptation cycle solution.....	78

Nomenclature

AOA – Angle of attack

AV - Artificial viscosity

B - Projection tensor

BC – Boundary conditions

CFL – Courant number

CV - Control volume

CS - Control surface

c - Speed of light in vacuum

c_p - Specific heat at constant pressure

c_v - Specific heat at constant volume

$\frac{D(\)}{Dt}$ - Substantial derivative

$\mathbf{E}, \mathbf{F}, \mathbf{G}$ - Viscous fluxes

E - System's energy

E_t - Total energy per unit volume

e - Internal energy per unit mass

FEM – Finite element method

FENSAP - Finite Element Navier-Stokes Analysis Package

F, G - Factors that divide R_m

$\mathbf{f}$ - Body forces per unit volume

GMRES – generalized minimum residual

H - Absolute value of the Hessian matrix

H - Stagnation enthalpy

h - Characteristic length of the element

$\tilde{k}$ - Numerical diffusivity

L - Length

M - Mach number

m - System's mass

N – Basis functions

$\mathbf{n}$ - Normal to the domain's frontier

nsd - Number of space dimensions

$\emptyset$ - Empty set

PDE – Partial differential equation

p - Pressure

Q - External heat per unit volume

$\mathbf{q}$ - Heat lost by conduction (per unit volume) through CS

R - Gas constant

Re – Reynolds number

R_m – Ratio of AV coefficient from momentum and continuity equations

R_p - Shock sensor

S – Shock sensor threshold

SU - Streamline Upwind

SUPG - Streamline Upwind Petrov-Galerkin

s - Entropy per unit mass

T - Temperature

t - Time

$\mathbf{U}$ - Conservation variables vector

u_i , $i = 1,2,3$ - Velocity components

$|u|$ - Absolute value of the local velocity

$\mathbf{V}$ - Velocity vector

$\|\mathbf{V}\|$ - Velocity's norm

x, y, z - Space coordinates

x_i , $i = 1,2,3$ - Spatial components

W - Weighting function

Γ - Domain's frontier

Ω - Space domain

γ - Ratio of specific heat
 δ_{ij} - Kroneker delta function
 Δx - Characteristic grid dimension
 μ - Coefficient of dynamic viscosity
 ∂ - Partial differential operator
 ε - Error in mesh adaptation theory
 κ - Coefficient of thermal conductivity
 λ - Second coefficient of viscosity
 ρ - Density
 ν - Eddy viscosity
 $v_{i,j}$, $v_{current\ node}$ - Shock sensor
 σ - Stress tensor
 σ_{ij} - Stress tensor
 τ_{ij} - Shear stress tensor
 τ - Shear stress tensor
 ∇ - Nabla differential operator

 ∞ - Free stream subscript
 o – Stagnation value subscript
 $\wedge$ - Denotes global derivative

1. Introduction

In the early 1950s when production fighter designs began pushing closer to the sound barrier, aircraft designers found that the drag on these aircraft increased substantially when the planes traveled near Mach 1, a phenomenon known as the transonic drag rise. This increase in drag is due to the formation of shock waves over portions of the airplane, which typically begins around Mach 0.8, and reaches a maximum at Mach 1. Because of its source, this type of drag is referred to as wave drag. The wave drag occurs in two ways. First, and primarily, the strong adverse pressure gradient across the shock causes the boundary layer to separate from the surface – this creates pressure drag due to flow separation. Second, even if the boundary layer did not separate, there is a loss of total pressure across the shock which ultimately would cause a net static pressure imbalance in the drag direction – also a pressure drag.

A 3D visualization of local supersonic pockets that could appear in transonic flight over various parts of the plane is contained in **Figure 1-1**. The Prandtl-Glauert condensation cloud [33], assumes a conical shape around the wings – the shock wave corresponds to the termination of the cloud, towards the aft of the plane, which gives rise to the characteristic flat base of the cone.

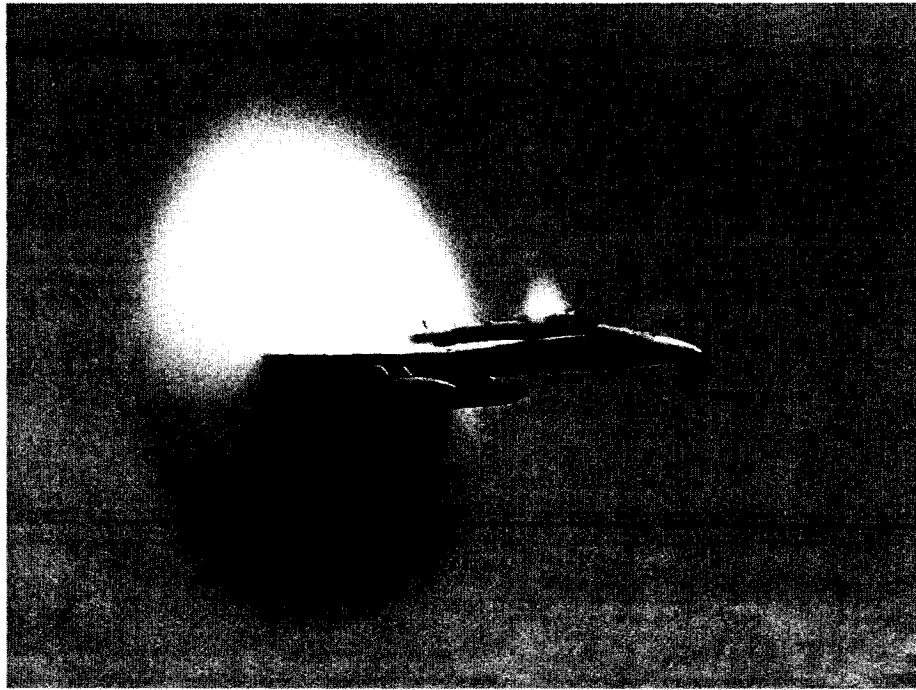


Figure 1-1: Prandtl-Glauert condensation cloud on a transonic F/A-18 Hornet plane (media release, USS Constellation, July 7, 1999 [990707-N-6483G-001])

No closed-form analytical formulas exist to predict the transonic drag rise. The prediction of transonic drag is so difficult that Jobe [18] in 1985 states: "The numerous authors in the field of numerical transonic aerodynamics have reached a consensus: Transonic drag predictions are currently unreliable by any method".

Advances in Computational Fluid Dynamics (CFD) during the last 40 years have made it possible to predict transonic drag. However, CFD is still very challenging, principally due to uncertainties in the calculation of the shock-induced separated flow.

Considerable effort has been devoted to the calculation of transonic viscous flows around airfoils, aircraft wings or complete aircraft using Finite Element Method (FEM), involving both structured and unstructured grids. The numerical simulations of compressible viscous flows results in a tightly coupled system of nonlinear equations in which parameters often take extreme values.

Such problems place special demands on the solution algorithm. Flow features such as boundary and shear layers, recirculation zones and shock

waves need to be accurately resolved to achieve an accurate solution. Furthermore, due to the mesh-dependent nature of CFD solutions, it is important to cluster grid points to regions of important flow physics. However, it is difficult to determine a priori where the mesh must be refined in order to accurately capture the physics of the flow.

FEM makes use of a spatial discretization and a weighted residual formulation to arrive at a system of matrix equations. The most common weighted residual formulation employed is the Galerkin method [6], in which weight and interpolation functions are identical. When applied to fluid flows or convective heat transfer, this method leads to a nonsymmetrical matrix associated with the convective terms, and as a consequence the solutions are often corrupted by spurious node-to-node oscillations. It is also known that the Galerkin FEM gives rise to central-difference type approximations of differential operators, leading to instability and upwind differencing must be used on the convective terms to obtain stable solutions. The drawback shown by this approach is that upwind differences are only first-order accurate, compared to central differences that are second-order accurate. Subsequently, it becomes apparent that a combination of central and upwind differences can improve solutions that employ either upwind or central differences alone. For a simple one dimensional model problem, it was possible to select the combination which resulted in exact nodal solutions. Equivalently, the proper amount of artificial diffusion could be added to the central formulation, procedure usually referred to as *optimal* or *smart* upwind method.

An initial upwind FEM was presented by Christie et al. [19] for the one dimensional advection-diffusion equation, by modifying weighting functions to achieve the upwind effect. In essence, the element upstream of a node is weighted more heavily than the element downstream of a node. This method was later generalized to the two dimensional case by Heinrich et al. [20]. Because the modified weighting function is applied to all terms in the equations, these formulations lead to consistent Petrov-Galerkin weighted residual methods.

Hughes [21] introduced a simple method for generating upwind elements, which made use of a modified quadrature rule for the convection term, while Hughes and Atkinson [22], using a different approach, derived an optimal upwind method from a variational principle, demonstrating that upwind methods may be developed from a firm theoretical basis. Many optimal upwind FEM give exact solution for the one dimensional problem, but when generalized to multidimensional flow situations some of these formulations are far from optimal, generating solutions that often exhibit excessive diffusion perpendicular to the flow direction. In addition, Galerkin formulation may provide in many instances oscillation free solutions which are more accurate than upwind solutions. To address criticism caused by the above results, Brooks et al.[13], introduced the Streamline Upwind Petrov-Galerkin (SUPG) formulation, where the added viscosity has an anisotropic character, acting just in the flow direction. This is achieved through the standard Galerkin weighting functions by adding a streamline upwind perturbation, which acts just in the flow direction. This method successfully incorporates streamline upwind concept, which precludes the possibility of excessive crosswind diffusion while eliminates artificial diffusion that plagues many classical upwind schemes by the consistent Petrov-Galerkin formulation. Additionally, the method is quite easy to implement and does not require the use of higher order weighting functions. Further investigations of SUPG method in the context of the multidimensional advection-diffusive equation were done by Johnson [23] and Nävert [24], who established optimal convergence rates and a strong discontinuity capturing property, even when the discontinuity is skew to the mesh. In Hughes et al. [25], the SUPG procedure of [13] is generalized to hyperbolic systems of conservation laws, with emphasis on high speed flows with shocks.

SUPG is an excellent method for problems with smooth solutions, but typically introduces localized oscillations about sharp internal and boundary layers. To improve upon the situation, Hughes et al. [30] added a discontinuity capturing term to the formulation, which provides additional control over gradients in the discrete solution and considerably increases the robustness of

the methodology. The developments were restricted to the scalar advection-diffusion equation. In Hughes et al. [31], the discontinuity capturing operator has been generalized to multidimensional systems.

It is well known that application of Galerkin FEM for fluid flow introduces numerical instabilities. First, the lack of a diffusive term in the continuity equation makes the advective-diffusive Navier-Stokes system of equations an incomplete parabolic one. The incompressible constraint leads to an indetermination of the system of governing equations since the unknown pressures have to be computed from the continuity equation. One method to solve the indetermination employs different function spaces for the velocity and pressure interpolation (e.g. quadratic- velocity and linear-pressure). This approach is known as Babuska-Brezzi stability condition [6], but in general it's not attractive from an implementation standpoint. Secondly, modeling non-symmetric advective terms by employing symmetric operators (such as centered finite differences or Galerkin basis functions) leads to solutions corrupted by spurious oscillations.

The Babuška-Brezzi stability condition, also known as inf-sup condition [26][27], are not easy to implement in practice, especially for three dimensional computations. However, N'Dri et al. [28] proposed a mixed space-time finite element formulation where the approximation spaces for velocity and pressure satisfy the LBB stability condition, showing that this formulation is stable and also applicable to transient viscous flows. A more common way of solving flow problems is by circumventing the Babuška-Brezzi criterion. A key idea is to treat Petrov-Galerkin formulations as devices to enhance stability without upsetting consistency. Hughes et al.[29] exploited this idea for Stokes flow.

Thus, the weighting function which multiplies the momentum-balance residual is not simply the Galerkin weight w^h , but w^h plus another term, sometimes referred to as the 'perturbation' of the Galerkin weighting function. This formulation may be considered a Petrov-Galerkin method and proved to be convergent for rather general C^0 combinations of velocity and pressure, in particular equal-order interpolations very attractive from a computational standpoint.

In a recent research [32], Almeida et al. introduced a Consistent Approximate Upwind Method (CAU). The idea is to keep the SUPG perturbation term over the streamline direction and add, in a consistent way, a non-linear perturbation to provide the control over the derivatives in the direction of the approximate gradient, avoiding completely spurious oscillations. This method is further combined with an h -adaptive mesh refinement procedure, as the finite element mesh near discontinuities in the flow must be fine enough in order to accurately solve all flow details.

1.1. Dissertation directions

This dissertation focuses on how to improve the capture of discontinuity of the compressible viscous Navier-Stokes (N-S) solution, such as a shock wave. We adopt a reliable shock detector formulation and apply it to the SUPG finite element formulation.

Our development is based on the success of our current capability to accurately compute the compressible viscous Navier-Stokes equations [9] as long as no strong discontinuity exists in the flow field. Basic FEM CFD formulation is demonstrated in Chapter 2. Our proposed shock detection can eliminate the limitation of the existing Navier-Stokes solver.

To demonstrate these benefits, we implement the shock detector including enhancement techniques such as mesh adaptation to the current N-S solver and apply it to transonic viscous flow past airfoils and wings. Chapter 3 contains the development of such detection and mesh adaptation. Comparisons with experimental data indicate good agreement for both 2D and 3D flows, as shown in Chapters 4 and 5. Chapter 6 summarizes this dissertation and suggests possible extension.

2. Governing Equations

In this chapter, a brief overview of the field equations and solution strategies are given. Also, the Finite Element Navier-Stokes Analysis Package (FENSAP), which was used as a Navier-Stokes solver for this work is introduced and a succinct description of software implementation and computation algorithms are added.

2.1. Navier Stokes equations in differential form

The fundamental equations that govern the fluid flow of a Newtonian fluid (i.e. stress varies linearly with strain rates) are obtained based on the following universal laws of conservation:

1. Conservation of mass: “matter can neither be created nor destroyed”.
2. Conservation of momentum: “the total force acting on a fluid particle is equal to the time rate of change of its momentum – Newton’s *lex secunda*”.
3. Conservation of energy: “energy can neither be created nor destroyed – First Law of Thermodynamics”

Note: According to Einstein theory, $E = m \cdot c^2$ (E stands for energy, m for mass, c for speed of light in vacuum) i.e. for a small variation in mass, a physical system can have an immense variation in energy. However, in fluid flow problems the energy variation does not generally exceed 10^5kJ/kg , so the variation of mass is practically negligible. For problems that do not contain nuclear reactions or other kind of transformations that allow for mass to be transformed into energy, the above conservation principles hold.

The derivation of the fundamental equations of fluid dynamics will not be presented here. A derivation based on the postulated relations between stress and rate of strain and heat flux and temperature gradient is treated for instance by Schlichting [8]. As we are going to use Navier–Stokes equations to obtain solutions for air flow around aerodynamic 2D and 3D objects, the assumptions of homogeneous and uniform fluid without mass diffusion or chemical reactions are made.

2.1.1. Continuity equation

Let's consider an infinitesimal control volume (CV) that is bounded by its control surface (CS). The Conservation of Mass law applied to a fluid that passes through CV yields the following equation of continuity:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \quad (2-1)$$

where ρ is fluid density and $\mathbf{V}$ is the fluid velocity. Using the formula for divergence applied to $\rho \mathbf{V}$ we get

$$\nabla \cdot (\rho \mathbf{V}) = \mathbf{V} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{V} \quad (2-2)$$

and the substantial derivative

$$\frac{D(\quad)}{Dt} = \frac{\partial(\quad)}{\partial t} + \mathbf{V} \cdot \nabla(\quad) \quad (2-3)$$

can now be used to write the continuity equation as:

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \mathbf{V}) = 0. \quad (2-4)$$

The above equation is obtained in the hypothesis that the control volume is fixed, and the changes to the fluid properties are done as the fluid flows through the CV. This approach is called Eulerian, by opposition to a Lagrangian approach, which considers the observer that records fluid properties moving together with the fluid element. The first approach is commonly used in fluid mechanics.

2.1.2. Momentum equations

Applying Newton's Second Law of dynamics for a fluid element passing through an infinitesimal and fixed CV, we get the following formulation of momentum equation:

$$\frac{\partial}{\partial t}(\rho \mathbf{V}) + \nabla \cdot \rho \mathbf{V} \mathbf{V} = \rho \mathbf{f} + \nabla \cdot \sigma_{ij} \quad (2-5)$$

The above formula contains, in the left hand side, the rate of increase of momentum per unit volume in the CV, the second term being the rate of

momentum lost by convection (per unit volume) through the CS. On the right hand side, the first term represents the body forces per unit volume, while the second term represents the surface forces per unit volume. On the left hand side, the divergence is applied to the tensor $\rho \mathbf{V}\mathbf{V}$, and can be expanded as:

$$\nabla \cdot \rho \mathbf{V}\mathbf{V} = \rho \mathbf{V} \cdot \nabla \mathbf{V} + \mathbf{V}(\nabla \cdot \rho \mathbf{V}) \quad (2-6)$$

Substituting equation (2-6) into equation (2-5), and simplifying the result through the continuity equation (2-4) we can write the momentum equation as:

$$\rho \frac{D\mathbf{V}}{Dt} = \rho \mathbf{f} + \nabla \cdot \boldsymbol{\sigma}_{ij} \quad (2-7)$$

In this equation, $\boldsymbol{\sigma}_{ij}$ represents the stress tensor, which contains the normal stresses and shearing stresses. For a Newtonian fluid, it's possible to derive a general deformation law that relates the stress tensor to the pressure and velocity components [8]. In compact tensor notation, expression of $\boldsymbol{\sigma}_{ij}$ becomes:

$$\sigma_{ij} = -p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \frac{\partial u_k}{\partial x_k} \quad i, j, k = 1, 2, 3 \quad (2-8)$$

where δ_{ij} is the Kroneker delta function ($\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$), u_1, u_2, u_3 represent the three components of the velocity vector $\mathbf{V}$, x_1, x_2, x_3 represent the three components of the position vector, μ is the coefficient of dynamic viscosity, and λ is the second coefficient of viscosity. After substituting the Stokes relation,

$$2\mu + 3\lambda = 0 \quad , \quad (2-9)$$

into equation (2-8) and taking into account eq. (2-3), we obtain the Navier-Stokes equation in differential form:

$$\rho \frac{D\mathbf{V}}{Dt} = \rho \mathbf{f} - \nabla p + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \mu \frac{\partial u_k}{\partial x_k} \right] \quad (2-10)$$

The Navier-Stokes equations form the basis upon which the entire science of viscous flow theory has been developed. Strictly speaking, the term Navier-Stokes equations refers to the components of viscous momentum equation (2-10). However, it is common practice to include the continuity equation and the

energy equation in the set of equations referred to as the Navier-Stokes equations.

2.1.3. Energy equation

Applying the First Law of thermodynamics for a fluid element passing through an infinitesimal and fixed CV, we get the following formulation of the energy equation:

$$\frac{\partial E_t}{\partial t} + \nabla \cdot E_t \mathbf{V} = \frac{\partial Q}{\partial t} - \nabla \cdot \mathbf{q} + \rho \mathbf{f} \cdot \mathbf{V} + \nabla \cdot (\sigma_{ij} \cdot \mathbf{V}) \quad (2-11)$$

where E_t is the total energy per unit volume given by

$$E_t = \rho \left(e + \frac{V^2}{2} + \text{potential energy} + \dots \right) \quad (2-12)$$

and e is the internal energy per unit mass. The first term of the left hand side of equation (2-11) represents the rate of increase of E_t in the CV, while the second term represents the rate of total energy lost by convection (per unit volume) through the CS. The first term on the right hand side of equation (2-11) is the rate of heat produced per unit volume by external agencies, while the second term is the rate of heat lost by conduction (per unit volume) through the CS. The third term on the right hand side of equation (2-11) represents the work done on the CV (per unit volume) by the body forces, while the fourth term represents the work done on the CV (per unit volume) by the surface forces. Fourier's law for the heat transfer will be assumed, so the heat transfer can be expressed as

$$\mathbf{q} = -\kappa \nabla T, \quad (2-13)$$

where k is the coefficient of thermal conductivity and T is the temperature.

2.2. Navier Stokes equations in conservation form

Conservation-law form is a convenient way of writing the field equations, and bellow we give this compact vector form of compressible Navier-Stokes equations in Cartesian coordinates, without body forces, mass diffusion, finite rate chemical reactions or external heat addition,[7].

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = 0, \quad (2-14)$$

where $\mathbf{U}$, $\mathbf{E}$, $\mathbf{F}$ and $\mathbf{G}$ are vectors given by:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{bmatrix}, \quad (2-15)$$

$$\mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ (E_t + p)u - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} + q_x \end{bmatrix}, \quad (2-16)$$

$$\mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ \rho vw - \tau_{yz} \\ (E_t + p)v - u\tau_{xy} - v\tau_{yy} - w\tau_{yz} + q_y \end{bmatrix}, \quad (2-17)$$

$$\mathbf{G} = \begin{bmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho w^2 + p - \tau_{zz} \\ (E_t + p)w - u\tau_{xz} - v\tau_{yz} - w\tau_{zz} + q_z \end{bmatrix}, \quad (2-18)$$

where the components of the stress tensor τ_{ij} are :

$$\tau_{xx} = \frac{2}{3} \mu \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right),$$

$$\tau_{yy} = \frac{2}{3} \mu \left(2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right),$$

$$\tau_{zz} = \frac{2}{3} \mu \left(2 \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right),$$

$$\begin{aligned}
\tau_{xy} &= \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) = \tau_{yx}, \\
\tau_{xz} &= \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) = \tau_{zx}, \\
\tau_{yz} &= \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) = \tau_{zy}.
\end{aligned} \tag{2-19}$$

For air in a laminar flow, the viscosity is defined empirically by Sutherland's law

$$\frac{\mu}{\mu_{\infty}} = \left(\frac{T}{T_{\infty}} \right)^{3/2} \cdot \left(\frac{T_{\infty} + 110}{T + 110} \right), \tag{2-20}$$

where T refers to the temperature in Kelvin, and variables noted with the subscript ∞ are reference values, [9]. The total or stagnation enthalpy is defined as

$$H = \frac{1}{2} V^2 + \frac{\gamma}{\gamma - 1} \frac{p}{\rho}, \tag{2-21}$$

where γ is the ratio of specific heat and equals 1.4 for air. The thermal conductivity k of formula equation (2-13) can be computed in a similar way as the laminar dynamic viscosity:

$$\frac{\kappa}{\kappa_{\infty}} = \left(\frac{T}{T_{\infty}} \right)^{3/2} \cdot \left(\frac{T_{\infty} + 133.7}{T + 133.7} \right). \tag{2-22}$$

The equation of state for an ideal gas, necessary to close the system of equations, is:

$$\frac{p}{\rho} = RT, \tag{2-23}$$

where R represents the gas constant.

2.3. Reference variables

In FENSAP, the non-dimensional variables were obtained with respect to four input parameters, namely reference pressure p_{∞} , temperature T_{∞} , length L_{∞} and the norm of the velocity vector V_{∞} . In the limit of ideal gas approximation, we can define the following non-dimensional quantities:

$$M_\infty = \frac{V_\infty}{\sqrt{\gamma R T_\infty}} , \quad (2-24)$$

$$\rho_\infty = \frac{P_\infty}{R T_\infty} , \quad (2-25)$$

$$H_\infty^* = \frac{\gamma}{\gamma-1} p_\infty^* + \frac{1}{2} , \quad (2-26)$$

$$\text{Re}_\infty = \frac{\rho_\infty L_\infty U_\infty}{\mu_\infty} , \quad (2-27)$$

where the asterisks in equation (2-26) indicates non-dimensional variable.

2.4. Spalart-Allmaras turbulence model

The turbulence model used to compute solutions of transonic flows in this work is the one-equation Spalart-Allmaras model. A detailed discussion and comparison about few turbulence models available, including Spalart-Allmaras is presented in reference [10]. The model solves a PDE over the whole field for a working variable $\tilde{\nu}$, from which the effective eddy viscosity ν_T is computed from a relation

$$\nu_T = \tilde{\nu} f_{\nu_1} . \quad (2-28)$$

The transport equation that gives $\tilde{\nu}$ is:

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1} \tilde{S} \tilde{\nu} + \frac{1}{\sigma \text{Re}_\infty} \frac{\partial}{\partial x_k} \left[(\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_k} + \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} \right] - c_{w1} f_w \frac{1}{\text{Re}_\infty} \left(\frac{\tilde{\nu}}{d} \right)^2 , \quad (2-29)$$

where

$$\tilde{S} = S + \frac{1}{\text{Re}_\infty} \frac{\tilde{\nu}}{\kappa^2 d^2} f_{\nu 2} , \quad S = \sqrt{2 \Omega_{ij} \Omega_{ij}} , \quad \Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) . \quad (2-30)$$

In the equation (2-30) d represents the distance to the wall, and $f_{\nu 2}$ and $f_{\nu 1}$ are defined as:

$$f_{\nu 1} = \frac{\chi^3}{\chi^3 + c_{\nu 1}^3} , \quad f_{\nu 2} = 1 - \frac{\chi}{1 + \chi f_{\nu 1}} , \quad (2-31)$$

where $\chi = \frac{\tilde{\nu}}{\nu}$ with ν being the laminar viscosity. The destruction term is formed with:

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{\frac{1}{6}}, g = r + c_{w2} (r^6 - r), r = \frac{\tilde{\nu}}{\kappa^2 d^2 \text{Re}_\infty S + \tilde{\nu} f_{v2}} \quad (2-32)$$

The closure coefficients of the model are:

$$c_{b1} = 0.1335, c_{b2} = 0.622, c_{v1} = 7.1, \sigma = \frac{2}{3} \quad (2-33)$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1 + c_{b2})}{\sigma}, c_{w2} = 0.3, c_{w3} = 2, \kappa = 0.41 \quad (2-34)$$

2.5. Boundary conditions (BC)

The Navier-Stokes system of equations, comprising a scalar continuity equation, three scalar momentum equations and one scalar energy equation is a non-linear, coupled, hybrid system, or elliptic-hyperbolic in space. For this kind of set of equations, no boundary conditions have been mathematically defined to date. For this reason, physical considerations will be used to define the boundary conditions.

Research done on solid wall viscous flows, revealed that the relative velocity between the solid wall and the flow is zero in the subsonic and low supersonic regimes. Also, the velocity component normal to the wall is zero. These two very important observations are called “no slip” and “no penetration”, respectively.

In CFD there are an inlet boundary and an exit boundary. At the inlet, a specified velocity profile is imposed, while for the exit boundary only the pressure variable (for subsonic and transonic) is imposed along the entire surface, meaning that the flow is fully developed or $\frac{\partial \mathbf{V}}{\partial \mathbf{n}} = 0$. At a plane of symmetry, the following

conditions apply:

$$\nabla p \cdot \vec{n} = 0; \quad \nabla \rho \cdot \vec{n} = 0; \quad \nabla(V_{t1}) \cdot \vec{n} = 0; \quad \nabla(V_{t2}) \cdot \vec{n} = 0; \quad (2-35)$$

where subscripts t1 and t2 indicate velocity tangential components.

2.6. Finite Element Method (FEM) formulation

The finite element method was initially used in the field of structural analysis, and the concept of “element” originates in the techniques used in stress calculations, where a continuous media was divided into many substructures of various shapes, analyzed separately and then re-assembled.

It is convenient at this point to write Navier-Stokes equations (2-4), (2-10) and (2-11) in index-free fashion. If the gradient of a scalar quantity φ is written as:

$$\nabla \varphi = \varphi_{,i} = \frac{\partial \varphi}{\partial x} \vec{i} + \frac{\partial \varphi}{\partial y} \vec{j} + \frac{\partial \varphi}{\partial z} \vec{k} \quad (2-36)$$

and the divergence of a vector quantity Φ is written as:

$$\nabla \Phi = \Phi_{i,i} = \frac{\partial \Phi_x}{\partial x} + \frac{\partial \Phi_y}{\partial y} + \frac{\partial \Phi_z}{\partial z} , \quad (2-37)$$

we can write then the Navier-Stokes equations:

$$(\rho \mathbf{u}_k)_{,k} = 0 \quad , \quad k = 1, nsd \quad \text{continuity eq.}, \quad (2-38)$$

$$\rho \mathbf{u}_j \mathbf{u}_{i,j} = \sigma_{ij,j} \quad , \quad i, j = 1, nsd \quad \text{momentum eq.}, \quad (2-39)$$

$$(\rho \mathbf{u}_l H)_{,l} = (\kappa T_{,m} + \mathbf{u}_n \tau_{nl})_{,l} \quad , \quad l, m, n = 1, nsd \quad \text{energy eq.}, \quad (2-40)$$

where “nsd” stands for “number of space dimensions”, $\mathbf{u}$ represents the velocity vector, κ represents the coefficient of thermal conductivity, T is temperature, H is enthalpy, ρ the density, σ the stress tensor and τ the shear stress tensor.

The field equations are written for a space domain, Ω , that in finite element analysis is subdivided into a number of elements of arbitrary shape and size, the restrictions being that the elements may not overlap, have to cover the whole computational domain and have to obey conformity condition, i.e. two elements must share a face or a node. Each element contains a number of points that are situated either on the sides or inside the element. At these points, the values of the unknowns and their derivatives have to be determined, and the total number of unknowns at the nodes, and their derivatives are called the degrees of freedom of the numerical problem. The field variables are approximated by linear combinations of known basis functions N (also known as

shape or interpolation functions) .We can therefore write for an approximate solution $\tilde{u}$ of $u(\mathbf{x})$:

$$\tilde{u}(\mathbf{x}) = \sum_i u_i N_i(\mathbf{x}), \quad (2-41)$$

where the summation extends over all nodes, and u_i represents the unknowns at the nodes. A common choice for the shape functions N are polynomials of different order within each element and zero outside the considered element, with the observation that the higher the order, the more computer power is needed. As a consequence of this property the shape functions satisfy to:

$$N_I^{(e)}(\mathbf{x}) = 0 \text{ if } \mathbf{x} \text{ not in element } (e) \quad (2-42)$$

and for any point $\mathbf{x}_J$ we have:

$$N_I^{(e)}(\mathbf{x}_J) = \delta_{IJ}, \text{ } \delta_{IJ} \text{ being Kroneker delta function} \quad (2-43)$$

Based on the requirement to represent exact a constant function $u(\mathbf{x}) = \text{constant}$ we can write another condition for N :

$$\sum_I N_I^{(e)}(\mathbf{x}) = 1, \text{ for all } \mathbf{x} \in (e) \quad (2-44)$$

As an example, in FENSAP the shape functions are chosen to be polynomials of first order, which means the first derivatives are constants, and the second derivatives and higher are zero. This choice has advantages and disadvantages: the simplicity of the first order polynomial offers economical use of available computer power for large industrial size flow problems, while some extra methods have to be used in order to deal with the satisfaction of stability condition, known as the Babuska-Brezzi condition, [6]. A complete discussion about elements and shape functions used in finite element analysis can be found in [11]. The method of weighted residual or weak formulation will be now introduced, and for detailed derivations please refer to [6], [12].

Starting from equations (2-38), (2-39), and (2-40) we multiply them by a given weighting function W and integrate across the entire computational domain in order to obtain the weak formulation of the conservation of mass, momentum and energy. Conservation of mass in weak formulation then reads:

$$\int_{\Omega} W^{cont}(\rho \mathbf{u}_k)_{,k} d\Omega = 0, \quad k = 1, nsd \quad (2-45)$$

and after applying Green-Gauss theorem to the above integral we get:

$$\int_{\Omega} W_{,k}^{cont}(\rho \mathbf{u}_k) d\Omega - \oint_{\Gamma} W^{cont} \rho \mathbf{u}_k \mathbf{n}_k d\Gamma = 0 \quad (2-46)$$

Similarly, the conservation of momentum in weak formulation reads:

$$\int_{\Omega} \mathbf{W}^{mom}(\rho \mathbf{u}_j \mathbf{u}_{i,j} - \sigma_{ij,j}) d\Omega = 0, \quad i, j = 1, nsd \quad (2-47)$$

and applying Green-Gauss we get:

$$\int_{\Omega} \mathbf{W}^{mom}(\rho \mathbf{u}_j \mathbf{u}_{i,j}) d\Omega + \int_{\Omega} \mathbf{W}_{,j}^{mom} \sigma_{ij} d\Omega - \oint_{\Gamma} \mathbf{W}^{mom} \sigma_{ij} \mathbf{n}_i d\Gamma = 0 \quad (2-48)$$

The conservation of energy in weak formulation reads:

$$\int_{\Omega} W^{en}(\rho \mathbf{u}_l H - \kappa T_{,m} - \mathbf{u}_n \tau_{nl})_{,l} d\Omega = 0, \quad m, n, l = 1, nsd \quad (2-49)$$

or after applying the divergence theorem:

$$\int_{\Omega} W_{,l}^{en}(\rho \mathbf{u}_l H - \kappa T_{,m} - \mathbf{u}_n \tau_{nl}) d\Omega - \oint_{\Gamma} W^{en}(\rho \mathbf{u}_l H - \kappa T_{,m} - \mathbf{u}_n \tau_{nl}) \mathbf{n}_l d\Gamma = 0 \quad (2-50)$$

The advantage of writing the equations (2-45), (2-47) and (2-49) using the divergence theorem is that the second order diffusion derivatives in the momentum and energy equations are now reduced to first order derivatives. That means the shape functions have to be differentiated just one time at the most, and this condition is exploited by FENSAP which uses first order, linear shape functions N .

2.7. Artificial viscosity (AV)

It's well known that the convective terms in Navier-Stokes equations generate non-physical oscillations in the solution, when symmetric operators like elementary Galerkin shape functions (or equivalently centered finite difference stencil) are used. Also, in FENSAP there is another source of this spurious behavior, called oscillations, and that's the use of the same order shape functions for pressure and velocity, contravening to Babuska-Brezzi condition. To stabilize the solution, artificial viscosity is added in the following form:

$$\text{Continuity Eq.} - \varepsilon_{cont} \left\{ \nabla \cdot (B \nabla p) + \nabla \cdot (B_{cross}^p \nabla p) \right\} \quad (2-51)$$

$$x \text{ Momentum Eq.} - \varepsilon_{mom} \left\{ \nabla \cdot (B \nabla u) + \nabla \cdot (B_{cross}^u \nabla u) \right\} \quad (2-52)$$

$$y \text{ Momentum Eq.} - \varepsilon_{mom} \left\{ \nabla \cdot (B \nabla v) + \nabla \cdot (B_{cross}^v \nabla v) \right\} \quad (2-53)$$

$$z \text{ Momentum Eq.} - \varepsilon_{mom} \left\{ \nabla \cdot (B \nabla w) + \nabla \cdot (B_{cross}^w \nabla w) \right\} \quad (2-54)$$

$$\text{Energy Eq.} - \varepsilon_{en} \left\{ \nabla \cdot (B \nabla H) + \nabla \cdot (B_{cross}^H \nabla H) \right\} \quad (2-55)$$

where B_{cross}^H and B represent the tensors of projection on gradient and streamline direction, respectively.

The above defined artificial viscosity (AV) is of anisotropic type. There were observations that good results are obtain, using the 1D flow analogy, if the AV is applied in the streamline direction. Another preferred direction for AV addition is the cross wind, or gradient direction. FENSAP uses a combination of the two, and in equations (2-51) to (2-55), the first term in the bracket represents AV in the streamline direction, while the second term represents the AV in the gradient of flow variable direction. A more detailed discussion will be carried out in the next chapter, where the Streamline Upwind (SU) and Streamline Upwind Petrov-Galerkin (SUPG) stabilisation methods are explained.

2.8. FENSAP – solving strategy

A set of nine flow equations (2-20), (2-21), (2-22), (2-23), (2-38), (2-39), (2-40), and nine unknowns ($\rho, p, T, H, \mathbf{u}_i, \mu, \kappa$) have been described in this chapter. When the governing equations contain algebraic expressions, this expression and one of its variables can be lagged in the solution process, [9]. The equation of state gives the temperature in function of the density and pressure, Sutherland's law relates the dynamic viscosity to the temperature, the thermal conductivity law relates the thermal conductivity to the temperature and the total enthalpy is expressed in function of pressure, density and velocity components. The temperature, dynamic viscosity, thermal conductivity and total enthalpy are lagged variables, and are therefore computed only after the density, velocity components and pressure have been solved. The governing flow system has

thus been reduced to five equation and five unknowns. A further simplification can be made in the case of adiabatic flows, because the energy equation is not solved any more, but the total enthalpy invariance along a streamline of the flow is used instead. The density is replaced in continuity and momentum equations based on equation (2-21), and now the flow system to be solved contains four PDEs and four unknowns $p, u_i, i=1,2,3$. As the equations to be solved contain non linear convective terms, the Lax-Milgram theorem that guarantees the uniqueness of the solution can not be applied to the system of equations. However, this system can be linearized, for example by Newton's method, and if the initial guess of the solution is good, the process will converge to the solution of the non-linear system, [9].

The discretized linear system of equations are solved iteratively based on the generalized minimum residual (GMRES) method.

3. Methodology

This chapter elaborates on the artificial viscosity schemes used to stabilize the FEM, on grids and on mesh adaptation algorithms used in this work. The discontinuities detection technique used to capture the shock waves in transonic viscous flows is presented and the numerical implementation within the commercial on-the-shelf software FENSAP is shown.

3.1. Streamline Upwind (SU) and SU Petrov-Galerkin (SUPG)

The stabilization techniques described herein are numerical tools able to correct for instabilities introduced by the finite element Galerkin formulation for Navier-Stokes equations. The Galerkin method, that has as a particularity the fact that the shape functions and weighting functions are similar, when applied to structures or heat conduction problems, gives birth to symmetric stiffness matrixes. In this case, it can be shown that the solution possesses the “best approximation” property, i.e. the difference between the finite element solution and the exact solution is minimized with respect to a certain norm. It’s largely because of this “best approximation” property that Galerkin method was applied with such a success in structural applications. However, for fluid flow problems, the Galerkin method encountered challenging problems. The cause is that in fluid flows and convection heat transfer, the matrix associated to convection terms is nonsymmetrical, and as a result the “best approximation” property is lost. In practice, the solutions are affected by oscillations that resemble low frequency noise – oscillations being most likely to appear in convection dominated flows (high Reynolds numbers) when a downstream boundary condition forces a rapid change in the solution, [12].

In FENSAP, Galerkin finite element method was implemented, and as a solution stabilization technique the SUPG formulation has been implemented.

3.1.1. The stabilization of one-dimensional convective dominated flow

For a simple 1D case we will show the numerical diffusivity that is able to stabilize the solution. We consider the convective 1D problem where velocity $\mathbf{u}$ transports a scalar φ through the computational domain $\Omega \subset R^1$ (just the transformations on the convective term will be shown):

$$\mathbf{u} \cdot \frac{\partial \varphi}{\partial x} \quad (3-1)$$

The stability condition generally requires that for each incorrect variation of the transported variable φ , the convective governing term should correct and compensate such an error. Lets consider now the central finite difference operator applied to equation (3-1), where the subscripts $(i-1)$ and $(i+1)$ indicate the grid points located upwind and downwind from the grid point i , respectively. We can write:

$$\mathbf{u} \cdot \frac{\partial \varphi}{\partial x} \approx \mathbf{u} \frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x} \quad (3-2)$$

where clearly appears that the sensitivity of the convective term with respect to the variation of scalar φ_i is zero. The conclusion is that modeling the convective terms of Navier-Stokes equations with spatially symmetrical operators leads towards neutral stability. The first attempt to recover the algorithm stability is the use of upwind finite difference schemes, so that the spatial derivatives of the variable in each nodal point (i) depends on the value assumed at the node itself (i) and on the value at the node at an upwind location. The directional stability is therefore recovered, on the expense of the formulation accuracy. An equivalent stabilisation can be obtained by mixing a convective term modeled with a centered finite difference stencil and an artificial diffusivity (viscosity) term:

$$\begin{cases} \mathbf{u} \cdot \frac{\partial \varphi}{\partial x} \approx \mathbf{u} \frac{\varphi_i - \varphi_{i-1}}{\Delta x} = \mathbf{u} \frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x} + \frac{\mathbf{u}\Delta x}{2} \cdot \frac{-\varphi_{i+1} + 2\varphi_i - \varphi_{i-1}}{\Delta x^2}, \text{ if } \mathbf{u} > 0 \\ \mathbf{u} \cdot \frac{\partial \varphi}{\partial x} \approx \mathbf{u} \frac{\varphi_i - \varphi_{i+1}}{\Delta x} = \mathbf{u} \frac{\varphi_{i-1} - \varphi_{i+1}}{2\Delta x} + \frac{\mathbf{u}\Delta x}{2} \cdot \frac{-\varphi_{i+1} + 2\varphi_i - \varphi_{i-1}}{\Delta x^2}, \text{ if } \mathbf{u} < 0 \end{cases}$$

The above two formulae can be combined in only one formulation:

$$\mathbf{u} \cdot \frac{\partial \varphi}{\partial x} \approx |\mathbf{u}| \frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x} + \frac{|\mathbf{u}|\Delta x}{2} \cdot \frac{-\varphi_{i+1} + 2\varphi_i - \varphi_{i-1}}{\Delta x^2}, \text{ for any } \mathbf{u} \quad (3-3)$$

where

$$\tilde{k} = \frac{|\mathbf{u}|\Delta x}{2}, \quad (3-4)$$

defines the numerical diffusivity that directly depends on the magnitude of convective phenomena and the characteristic grid dimension Δx , and $|\mathbf{u}|$ is the absolute value of the local velocity. The interpretation of this result is that stabilized schemes could be obtained using centered differencing and artificial diffusivity approach as in equation (3-3), and the goal is to correct the under diffusivity of the Galerkin scheme using the above described idea.

3.1.2. Upwind and Streamline Upwind stabilisation

The Galerkin formulation can be stabilized using the addition of artificial balancing integral able to compensate the negative diffusivity of the weighted residual method. An appropriate choice of the artificial contribution intensity could lead, for a 1D case, to the exact nodal solution – such upwind schemes are called optimal or smart, [6]. The artificial diffusivity in the *optimal upwind scalar schemes* could be defined as:

$$\tilde{k} = \left(\frac{|u|h}{2} \right) \zeta \quad (3-5)$$

$$\zeta = \coth(\alpha) - \frac{1}{\alpha} \quad (3-6)$$

$$\alpha = \frac{|u|h}{2k} \quad (3-7)$$

where :

h , is the characteristic length of the element;

α , is the elementary grid Peclet number;

k , is the physical fluid diffusivity

ζ , is a stabilisation coefficient able to modify the intensity of artificial diffusivity with reference to the magnitude of convection transport phenomena.

The application of *upwind scalar schemes* to multidimensional flow conditions leads often to non controlled over-diffusion solutions, in particular showing non physical cross wind diffusion, because of the isotropic character of the balancing operator.

In order to eliminate the drawbacks shown by the scalar schemes, the *streamline upwind* technique has been developed, where the upwind effect is concentrated in principle along the streamline direction. In such a method, the balancing operator, in the form of a diffusive term, acts exclusively in the streamline direction as an anisotropic artificial diffusivity. The artificial diffusivity assumes therefore a tensorial character and could be expressed as follows:

$$\tilde{k}_{ij} = \tilde{k} \cdot \bar{u}_i \bar{u}_j, \quad (3-8)$$

where:

$\bar{u}_j = \mathbf{u}_j / \|\mathbf{u}\|$, defines the velocity components unit vector.

$\|\mathbf{u}\| = \sqrt{\mathbf{u}_i \mathbf{u}_i}$, defines the velocity norm.

$\tilde{k}$, represents the artificial diffusivity already defined with reference to the scalar upwind technique.

Let's look now at the form of the tensorial balancing term. The divergence of the symmetric part of the Cauchy stress tensor that appears in the molecular diffusive term can be written as:

$$(k_{ij}(\mathbf{u}_{i,j} + \mathbf{u}_{j,i}))_{,j} = (k_{ij}(\mathbf{u}_{i,j}))_{,j} + (k_{ij}(\mathbf{u}_{j,i}))_{,j} \quad (3-9)$$

and reversing the derivation order on the second term we get:

$$(k_{ij}(\mathbf{u}_{i,j}))_{,j} + (k_{ij}(\mathbf{u}_{j,i}))_{,j} = (k_{ij}(\mathbf{u}_{i,j}))_{,j} + (k_{ij}(\mathbf{u}_{j,j}))_{,i}. \quad (3-10)$$

The diffusive integral term in Galerkin residual formulation (equation (2-48)) can be written as:

$$\int_{\Omega} \mathbf{W}_{,i}^{mom} \sigma_{ij} d\Omega = \int_{\Omega} \mathbf{W}_{,i}^{mom} (k_{ij} + \tilde{k}_{ij}) \mathbf{u}_{i,j} d\Omega \quad (3-11)$$

where the diffusivity explicitly appears as the sum of physical and artificial contribution:

$$\int_{\Omega} \mathbf{W}_{,i}^{mom} (k_{ij}) \mathbf{u}_{i,j} d\Omega + \int_{\Omega} \mathbf{W}_{,i}^{mom} (\tilde{k}_{ij}) \mathbf{u}_{i,j} d\Omega \quad (3-12)$$

Now we can introduce in the second term of expression (3-12) the artificial diffusivity expression of (3-8) and we get the expression of the stabilisation integral as:

$$\int_{\Omega} \mathbf{W}_{,i}^{mom} \tilde{k} \bar{\mathbf{u}}_i \bar{\mathbf{u}}_j \mathbf{u}_{i,j} d\Omega . \quad (3-13)$$

Substituting the versor $\bar{\mathbf{u}}_j$ in accordance with its definition, it is possible to write:

$$\int_{\Omega} \mathbf{W}_{,i}^{mom} \frac{\bar{\mathbf{u}}_i}{\|\mathbf{u}\|} \tilde{k} \mathbf{u}_j \mathbf{u}_{i,j} d\Omega \quad (3-14)$$

From the above equation (3-14) clearly appears that the tensorial stabilisation term has the form of a convective integral that must be controlled.

3.1.3. Streamline Upwind – Petrov Galerkin stabilisation

The consistency of the stabilisation methods could be recovered extending the weights perturbation, limited to the convective integral in the *streamline upwind* scheme, to each term that is contained in the residual Navier-Stokes formulation (2-48). In such a way, the built residual structure assumes the character of a Petrov-Galerkin formulation, due to the introduction of different function spaces used for the approximation of the solution and of the variations (weights). Such a residual stabilized finite element formulation for convection dominated flows is called *streamline upwind – Petrov Galerkin* (SUPG) , [12].

Lets now consider a flow domain $\Omega \in R^{nsd}$ which has a boundary Γ defined by piecewise continuous functions. Consider also a point $x_i (i = 1, nsd)$ belonging to Ω , and $\mathbf{n}_i$ as the component along i direction of the normal unit vector to Γ (positive direction toward the inner of the domain). The domain boundary Γ is split into two subsets Γ_g and Γ_h that satisfy the following relations:

$$\Gamma_g \cup \Gamma_h = \Gamma \quad (3-15)$$

$$\Gamma_g \cap \Gamma_h = \emptyset \quad (3-16)$$

The domain Ω is subdivided into a finite number of elements $\Omega_e, e=1, \text{numel}$ where numel represents the total number of elements in the domain. We denote by Γ_e the boundary of the element Ω_e , and the following relations are true:

$$\cup_e \Omega_e = \Omega \quad (3-17)$$

$$\cap_e \Omega_e = \emptyset \quad (3-18)$$

The interior boundary Γ_{int} is defined as:

$$\Gamma_{\text{int}} = \cup_e \Gamma_e - \Gamma \quad (3-19)$$

Recall the PDE that governs the steady compressible flow:

$$\rho \mathbf{u}_j \mathbf{u}_{i,j} - \sigma_{ij,j} = 0 \quad (3-20)$$

$$(\rho \mathbf{u}_i)_{,i} = 0 \quad (3-21)$$

with the following set of boundary conditions:

$$\begin{aligned} \mathbf{u}_i &= g_i \longrightarrow \Gamma_g, \text{ essential Dirichlet BC} \\ \sigma_{ij} \mathbf{n}_i &= h_j \longrightarrow \Gamma_h, \text{ natural Neumann BC} \end{aligned} \quad (3-22)$$

As the classical Galerkin method uses identical collections of trial and weighting functions, the weights are continuous across the inter-element boundaries. Such a property is lost in the case of SUPG formulation, due to the set of weights that modify the original Galerkin functions on an element basis as follows:

$$\mathbf{W}^{mom} = \mathbf{W}^{mom} + \mathbf{p}^{mom} \quad (3-23)$$

where $\mathbf{W}^{mom}$ is the Galerkin weight applied to the momentum equation and $\mathbf{p}^{mom}$ is the stabilising *streamline upwind* like contribution. The application of such a perturbation introduces the discontinuity of the above mentioned weighting function, but the perturbation function still has the integrability property on the element scale. Let's consider a point x belonging to the interior boundary Γ_{int} and arbitrarily establish a positive orientation for the normal direction across the boundary. We define $\mathbf{n}^+$ and $\mathbf{n}^-$ as the unit vectors normal to Γ_{int} in the considered nodal position x . The relation between the two unit normal vectors is expressed by:

$$\mathbf{n}^+ = -\mathbf{n}^- \quad (3-24)$$

Introducing for simplicity a term that accounts for the sum of convection and diffusive fluxes:

$$\chi_i = \chi_i^a + \chi_i^d = \rho \mathbf{u}_j \mathbf{u}_{i,j} - \sigma_{ij,j} \quad (3-25)$$

It's possible to show that the jump across the neighbouring elements of χ_i at the considered nodal point x , defined as:

$$[\chi_n] = (\chi_i^+ - \chi_i^-) \mathbf{n}_i^+ = \chi_i^+ \mathbf{n}_i^+ + \chi_i^- \mathbf{n}_i^- \quad (3-26)$$

is an invariant with respect to the adopted sign convention for Γ_{int} .

On the basis of above mentioned perturbation, the SUPG method applied to Navier-Stokes equations leads to the following residual stabilized formulation:

$$\begin{aligned} \sum_e \int_{\Omega_e} [\mathbf{W}^{mom} (\rho \mathbf{u}_j \mathbf{u}_{i,j}) + \mathbf{W}_{,i}^{mom} \sigma_{ij}] d\Omega + \sum_e \int_{\Omega_e} [\mathbf{p}^{mom} (\rho \mathbf{u}_j \mathbf{u}_{i,j}) - \mathbf{p}^{mom} \sigma_{ij,j}] d\Omega - \\ - \sum_e \int_{\Gamma_{\text{int}}} \mathbf{W}^{mom} [\sigma_{ij} \mathbf{n}_i] d\Gamma - \int_{\Gamma_h} \mathbf{W}_h h_i d\Gamma = 0 \end{aligned} \quad (3-27)$$

where $\mathbf{W}_h$ is the restriction of weight function on the boundary of the computational domain.

An equivalent way of writing (3-27) is:

$$\begin{aligned} \sum_e \int_{\Omega_e} \mathbf{W}^{mom} [(\rho \mathbf{u}_j \mathbf{u}_{i,j}) - \sigma_{ij,j}] d\Omega - \sum_e \int_{\Gamma_{\text{int}}} \mathbf{W}^{mom} [\sigma_{ij} \mathbf{n}_i] d\Gamma - \\ - \int_{\Gamma_h} \mathbf{W}_h (\sigma_{ij} \mathbf{n}_i - \mathbf{h}_i) d\Gamma = 0 \end{aligned} \quad (3-28)$$

It is worth to observe that from the obtained residual form of the integral problem is possible to extract original differential expression of the Navier-Stokes boundary problem.

The fundamental aspect that characterizes the obtained stabilized finite element formulation in comparison to the classical upwind schemes is that the *streamline upwind* perturbation function $\mathbf{p}^{mom}$ plays its role at the interior of the element where the perturbation itself is continuous.

The expression of *streamline upwind* perturbation of weights $\mathbf{p}^{mom}$ is defined starting from the modification of the convective weight obtained for the *streamline upwind* method (3-14). The equivalence between the modification of

the convective weight and the introduction of the tensorial diffusive balancing integral permits us to write the following, [6]:

$$\mathbf{p}^{mom} = \tilde{k} \bar{\mathbf{u}}_j \mathbf{W}_{,j}^{mom} / \|\mathbf{u}\| \quad (3-29)$$

The discontinuous effect introduced by the perturbation $\mathbf{p}^{mom}$ can be seen in **Figure 3-1**, where three consecutive nodes in an one dimensional case were used, the flow direction was indicated with the arrow and the weights for Petrov Galerkin scheme were drawn in dotted line. The fact that the slopes of the shape functions are different at node A determines the weights discontinuity as shown. FENSAP uses shape function of first order, therefore the derivatives are constants. It is interesting to note that in this case, the divergence of the diffusive fluxes is equal to zero

$$\sigma_{ij,j} = 0 \quad (3-30)$$

and the function $\mathbf{p}^{mom}$ does not play any stabilisation effect on the diffusive integral. We can say that in this case SUPG formulation becomes SU formulation.

The coefficient $\tilde{k}$ for multidimensional computations has the form:

$$\tilde{k} = \frac{(\xi \mathbf{u}_\xi \mathbf{h}_\xi + \eta \mathbf{u}_\eta \mathbf{h}_\eta + \zeta \mathbf{u}_\zeta \mathbf{h}_\zeta)}{2} \quad (3-31)$$

where:

$$\xi = \coth(\alpha_\xi) - \frac{1}{\alpha_\xi}$$

$$\eta = \coth(\alpha_\eta) - \frac{1}{\alpha_\eta}$$

$$\zeta = \coth(\alpha_\zeta) - \frac{1}{\alpha_\zeta}$$

$$\alpha_\xi = \frac{|\mathbf{u}_\xi| |\mathbf{h}_\xi|}{2k}$$

$$\alpha_\eta = \frac{|\mathbf{u}_\eta| |\mathbf{h}_\eta|}{2k}$$

$$\alpha_\zeta = \frac{|\mathbf{u}_\zeta| |\mathbf{h}_\zeta|}{2k}$$

$$\mathbf{u}_\xi = \mathbf{e}_\xi \cdot \mathbf{u}, \quad \mathbf{u}_\eta = \mathbf{e}_\eta \cdot \mathbf{u}, \quad \mathbf{u}_\zeta = \mathbf{e}_\zeta \cdot \mathbf{u} \quad (3-32)$$

In the above formulae $\mathbf{e}_\xi$, $\mathbf{e}_\eta$ and $\mathbf{e}_\zeta$ are the unit vectors of the computational frame of reference that has the directions ξ, η, ζ while $\mathbf{h}_\xi, \mathbf{h}_\eta, \mathbf{h}_\zeta$ are the corresponding elementary length scales.

3.2. Shock wave detector definition

Jameson et al. [14] used a finite volume discretization in conjunction with carefully designed dissipative terms of third order to efficiently solve the Euler equations for transonic regime in arbitrary domains. The shock sensor defined in [14] (see equation 3-37 in following section) was used to adaptively construct the dissipative terms blending the second and fourth differences.

Based on the formulation of equation 3-37, we constructed the shock sensor for the finite element method treated in the present work (see equation 3-41). While in the finite volume scheme the sensor is calculated for each cell, in the finite element method the sensor is calculated for each node of the computational domain.

3.2.1. Shock detector for 2D inviscid flow

To prevent the tendency of odd and even point decoupling, or checker board situation, and to prevent the appearance of oscillations in regions containing severe pressure gradients in the neighborhood of shock waves or stagnation points, it proves necessary to augment the finite volume scheme by the addition of artificial dissipative terms [1] [14]. The governing equation in conservation form will be (similar to equation (2-14), after neglecting viscous terms):

$$\frac{d}{dt}(hw) + Qw - Dw = 0 \quad (3-33)$$

where h is the cell area, Q is the spatial discretization operator, D is the dissipative operator and w represents the dependent variable. A sketch of the

discretization is given in **Figure 3-2** [14]. Extensive numerical experiments [14] have established that an effective form for D_w is a blend of second and fourth order differences with coefficients which depend on the local pressure gradient. The construction of the dissipative terms for each of the four dependent variables is similar. For the density we have:

$$D\rho = D_x\rho + D_y\rho \quad (3-34)$$

where $D_x\rho$ and $D_y\rho$ are the corresponding contributions for the two directions, written in conservation form:

$$\begin{aligned} D_x\rho &= d_{i+\frac{1}{2},j} - d_{i-\frac{1}{2},j} \\ D_y\rho &= d_{i,j+\frac{1}{2}} - d_{i,j-\frac{1}{2}} \end{aligned} \quad (3-35)$$

The terms on the right all have a similar form:

$$d_{i+\frac{1}{2},j} = \frac{h_{i+\frac{1}{2},j}}{\Delta t} \left\{ \varepsilon_{i+\frac{1}{2},j}^{(2)} (\rho_{i+1,j} - \rho_{i,j}) - \varepsilon_{i+\frac{1}{2},j}^{(4)} (\rho_{i+2,j} - 3\rho_{i+1,j} + 3\rho_{i,j} - \rho_{i-1,j}) \right\} \quad (3-36)$$

where $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ are adapted to the flow. Now, we define the shock sensor $v_{i,j}$ as a normalized second order difference of pressure [14]:

$$v_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{|p_{i+1,j}| + 2|p_{i,j}| + |p_{i-1,j}|} \quad (3-37)$$

Then,

$$\varepsilon_{i+\frac{1}{2},j}^{(2)} = \kappa^{(2)} \max(v_{i+1,j}, v_{i,j}) \quad (3-38)$$

$$\varepsilon_{i+\frac{1}{2},j}^{(4)} = \max\left(0, \left(\kappa^{(4)} - \varepsilon_{i+\frac{1}{2},j}^{(2)}\right)\right) \quad (3-39)$$

where typical values for constants $\kappa^{(2)}$ and $\kappa^{(4)}$ are [14]:

$$\kappa^{(2)} = \frac{1}{4}, \kappa^{(4)} = \frac{1}{256} \quad (3-40)$$

It has been found [14] that in smooth regions of the flow, the scheme is not sufficiently dissipative unless the fourth differences are included, while near the shocks it has been found that the fourth differences tend to introduce

overshoots, and therefore they are switched off by subtracting $\varepsilon^{(2)}$ from $\kappa^{(4)}$ in equation (3-39). The shock sensor has values of the order of magnitude 10^{-5} , if the flow does not show any discontinuities, like shock waves, and its value increases by several orders of magnitudes in the vicinity of afore mentioned discontinuities.

3.2.2. Shock Sensor implemented in FENSAP

As in this work the domain discretization is of unstructured type, the mesh in discussion will be unstructured, three-dimensional one. A representation of a 2D unstructured mesh is given in **Figure 3-3**, where the elements are represented by triangles. For simplicity, we are going to use this diagram to explain the local and global derivatives of a flow property, as well as the discontinuity detection mechanism. The discussion will also apply to a 3D mesh with tetrahedron cells. The shock sensor defined by equation (3-37) is for a structured mesh, as in **Figure 3-2**. For an unstructured mesh, the formula (3-37) has to be slightly modified. To compute the shock sensor for node "0" in **Figure 3-3**, we have to determine all the elements that have node "0" in common, and then calculate the minimum and maximum pressure for all nodes that are related to node "0" through a cell. These pressures are denoted by $p_{\max}$ and $p_{\min}$, and together with the pressure at the node "0" constitute the ingredients used to compute the shock sensor value at the node "0" under discussion. The same procedure is to be followed for all nodes in the mesh, and we end up with a shock sensor vector that has the length in accordance with the number of the nodes in the mesh. The formula used to compute the sensor is given below:

$$v_{current\ node} = \frac{|p_{\max} - 2 \cdot p_{current\ node} + p_{\min}|}{p_{\max} + 2 \cdot p_{current\ node} + p_{\min}} \quad (3-41)$$

Another particularity of FENSAP is the use of local and global derivatives. The local derivative is denoted by the Greek letter ψ while the global derivative is denoted by the same letter, but with a hat $\hat{\psi}$. The presence of these two types

of derivatives is going to be used to form *first* and *second order* artificial viscosity schemes, as will be explained later in this section.

To explain the idea of local and global derivatives, we can employ, again for simplicity, a 1D flow situation. Let's consider the scalar flow property φ in three consecutive nodes (i) , $(i-1)$ and $(i+1)$. The spatial discretization is not uniform, so from node $(i-1)$ to node (i) and node (i) to node $(i+1)$ we have distances h_{i-1} and h_i , respectively. The situation is shown in **Figure 3-4**.

Computing the derivatives of φ in all nodes will have to take into account that the property φ itself is continuous, but not derivable in the nodes. We can only say that the slope of φ from node $i-1$ to i is α_{i-1} and from node i to $i+1$ be α_{i+1} . In node i , for example, we can only take the approximation of the slope based on the values α_{i-1} and α_{i+1} previously computed:

$$\begin{aligned}\alpha_i &= \frac{\varphi_{i+1} - \varphi_{i-1}}{h_{i-1} + h_i} = \frac{\varphi_i - \varphi_{i-1}}{h_{i-1}} \cdot \frac{h_{i-1}}{h_{i-1} + h_i} + \frac{\varphi_{i+1} - \varphi_i}{h_i} \cdot \frac{h_i}{h_{i-1} + h_i} = \\ &= \alpha_{i-1} \cdot \frac{h_{i-1}}{h_{i-1} + h_i} + \alpha_{i+1} \cdot \frac{h_i}{h_{i-1} + h_i}\end{aligned}\tag{3-42}$$

where $\alpha_{i-1} = \frac{\varphi_i - \varphi_{i-1}}{h_{i-1}}$ and $\alpha_{i+1} = \frac{\varphi_{i+1} - \varphi_i}{h_i}$. This expression (3-42) represents a weighted summation of derivatives just before and just after the current node. The above discussion about local and global derivatives extends to 2D and 3D cases as well, and this concludes the global and local derivatives used in FENSAP.

3.2.3. Entropy based shock detection

A plane progressive perturbation in a fluid was observed to transform, in time, into a discontinuity surface for velocity, pressure, density, temperature. As we are interested to detect, for a steady flow, the zones that contain such a jump in above mentioned flow variables, the shock detector formula (3-41) has been chosen. The main ingredient used in (3-41) is the fluid pressure calculated in all

nodes of the discretized domain. Apart from pressure, we will also investigate if entropy would be a suitable parameter. From the definition, we can write [15] :

$$s - s_0 = c_p \ln \frac{T}{T_0} - R \ln \frac{p}{p_0} \quad (3-43)$$

where the function s represents the entropy per unit mass at a given state, p and T are pressure and temperature, respectively, at the same state, c_p represents the specific heat at constant pressure, R the gas constant while the subscript 0 indicates the corresponding initial state values of the variables.

Relation (3-43) can also be written as:

$$s - s_0 = c_p \ln \left[\frac{T}{T_0} \left(\frac{p_0}{p} \right)^{\frac{R}{c_p}} \right], \quad (3-44)$$

where the specific heats at constant pressure and constant volume have the expressions:

$$c_v = \frac{R}{\gamma - 1}, \quad c_p = \frac{\gamma \cdot R}{\gamma - 1}, \quad \gamma = 1.4. \quad (3-45)$$

From the equation of state (2-23) we can obtain a more useful expression for the entropy change:

$$s - s_0 = \gamma \cdot c_v \cdot \ln \left[\frac{\rho_0}{\rho} \left(\frac{p}{p_0} \right)^{\frac{1}{\gamma}} \right] = c_v \cdot \ln \left[\left(\frac{\rho_0}{\rho} \right)^{\gamma} \frac{p}{p_0} \right] = c_v \cdot \left(\ln \frac{p}{\rho^{\gamma}} - \ln \frac{p_0}{\rho_0^{\gamma}} \right) \quad (3-46)$$

As far as we are concerned, the initial entropy serves just as a reference, and we will only be interested in the variation of entropy. Therefore, from equation (3-

46) -last expression, we retain for sensor definition just the term $\frac{p}{\rho^{\gamma}}$ that is

responsible for variation. We obtain the nodal value of the sensor in the form:

$$v_{current\ node} = \frac{\left| \left(\frac{p}{\rho^{\gamma}} \right)_{\max} - 2 \cdot \left(\frac{p}{\rho^{\gamma}} \right)_{current\ node} + \left(\frac{p}{\rho^{\gamma}} \right)_{\min} \right|}{\left(\frac{p}{\rho^{\gamma}} \right)_{\max} + 2 \cdot \left(\frac{p}{\rho^{\gamma}} \right)_{current\ node} + \left(\frac{p}{\rho^{\gamma}} \right)_{\min}} \quad (3-47)$$

3.3. Streamline and Cross Wind directions definition

As was discussed before, in order to obtain valid, oscillations free solutions of the Navier-Stokes equation, artificial viscosity (AV) has to be used. Through extensive analysis and numerical experiments [6] [13], it was determined that AV has to be applied anisotropic in order to compensate for low diffusivity, as in the continuity equation case, or for instabilities generated by convection terms, as in momentum or energy equations. In FENSAP, AV is added in the streamline and cross wind directions, as will be explained further.

Considering the vectors $\mathbf{V}(V_1, V_2, V_3)$ and $\mathbf{a}(a_1, a_2, a_3)$ (see **Figure 3-5**), we can project $\mathbf{a}$ on $\mathbf{V}$ as follows:

$$\mathbf{a} \cdot \mathbf{V} = \mathbf{V} \cdot \mathbf{a} = a_1 \cdot \frac{V_1}{\|\mathbf{V}\|} + a_2 \cdot \frac{V_2}{\|\mathbf{V}\|} + a_3 \cdot \frac{V_3}{\|\mathbf{V}\|} \quad (3-48)$$

where $\mathbf{V}$ was divided by its norm, in order to transform it into an unitary vector. Now, we want the projection to be in the direction of velocity, so we multiply (3-48) by the velocity versor as follow:

$$\begin{aligned} \text{Projection in } \mathbf{V} \text{ direction} &= \left(a_1 \cdot \frac{V_1}{\|\mathbf{V}\|} + a_2 \cdot \frac{V_2}{\|\mathbf{V}\|} + a_3 \cdot \frac{V_3}{\|\mathbf{V}\|} \right) \cdot \frac{\mathbf{V}}{\|\mathbf{V}\|} = \\ &= (a_1 V_1 + a_2 V_2 + a_3 V_3) \frac{V_1 \mathbf{i} + V_2 \mathbf{j} + V_3 \mathbf{k}}{\|\mathbf{V}\|^2} = (a_1 V_1^2 + a_2 V_1 V_2 + a_3 V_1 V_3) \frac{1}{\|\mathbf{V}\|^2} \mathbf{i} + \\ &+ (a_1 V_1 V_2 + a_2 V_2^2 + a_3 V_2 V_3) \frac{1}{\|\mathbf{V}\|^2} \mathbf{j} + (a_1 V_1 V_3 + a_2 V_2 V_3 + a_3 V_3^2) \frac{1}{\|\mathbf{V}\|^2} \mathbf{k} = \\ &= \frac{1}{\|\mathbf{V}\|^2} \begin{bmatrix} V_1^2 & V_1 V_2 & V_1 V_3 \\ V_1 V_2 & V_2^2 & V_2 V_3 \\ V_1 V_3 & V_2 V_3 & V_3^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \mathbf{B} \mathbf{a} \end{aligned} \quad (3-49)$$

We notice the form of $\mathbf{B}$, and the tensor operation between $\mathbf{B}$ and $\mathbf{a}$ in order to obtain the desired projection. For the cross wind direction projection, the role of $\mathbf{V}$ is taken by $\boldsymbol{\omega}$, and the projection tensor $\mathbf{B}$ changes accordingly. Looking back at the definition of artificial viscosity formulae (2-51) to (2-55) and having in mind the already defined local and global derivatives, for the pressure, for example, we can write the expression of tensor $\mathbf{B}_{cross}^p$ as:

$$\mathbf{B}_{cross}^p = \frac{1}{\|\nabla \hat{p}\|^2} \begin{bmatrix} (\nabla \hat{p})_1^2 & (\nabla \hat{p})_1(\nabla \hat{p})_2 & (\nabla \hat{p})_1(\nabla \hat{p})_3 \\ (\nabla \hat{p})_1(\nabla \hat{p})_2 & (\nabla \hat{p})_2^2 & (\nabla \hat{p})_2(\nabla \hat{p})_3 \\ (\nabla \hat{p})_1(\nabla \hat{p})_3 & (\nabla \hat{p})_2(\nabla \hat{p})_3 & (\nabla \hat{p})_3^2 \end{bmatrix} \quad (3-50)$$

that means the local gradient of pressure is projected onto the global (or hat) gradient of pressure. A similar procedure will be applied to the other flow properties, and $\mathbf{B}_{cross}^u$, $\mathbf{B}_{cross}^v$, $\mathbf{B}_{cross}^w$, $\mathbf{B}_{cross}^H$ are now defined.

It is to be noted that differences between the local and global derivatives (or gradients, as we operate in a three dimensional space) are to be seen just if there is a variation of the flow property. Otherwise the gradient direction and magnitude in the two situations will be identical.

3.4. Artificial Viscosity Switching Scheme

Artificial viscosity is used in FENSAP as a stabilization method for the field equations, as briefly indicated in the previous section. We want now to explain what *first order AV* and *second order AV* mean, as we will explain later the AV switching procedure based on the flow behavior. As we know, the shape functions used here are of the first order type, which means their variation is linear, the first derivative is a constant, and higher order derivatives are zero. Based on this, the *first order AV* could be synthesized by the expression:

$$\int_{\Omega} \tilde{k}_{ij} \frac{\partial u_i}{\partial x_j} \frac{\partial W}{\partial x_j} d\Omega \quad (3-51)$$

where $\tilde{k}_{ij}$ is defined by equation (3-8) and W is the weight function. Similarly, using the global derivative this time, the following stabilisation integral can be defined:

$$\int_{\Omega} \tilde{k}_{ij} \frac{\partial \hat{u}_i}{\partial x_j} \frac{\partial W}{\partial x_j} d\Omega \quad (3-52)$$

We can attempt now to define the FENSAP's *second order AV* as the difference between two first order terms defined by equation (3-51) and equation (3-52):

$$-\underbrace{\int_{\Omega} \tilde{k}_{ij} \frac{\partial u_i}{\partial x_j} \frac{\partial W}{\partial x_j} d\Omega}_a + R_p \underbrace{\int_{\Omega} \tilde{k}_{ij} \frac{\partial \hat{u}_i}{\partial x_j} \frac{\partial W}{\partial x_j} d\Omega}_b = -\int_{\Omega} \tilde{k}_{ij} \left(\frac{\partial u_i}{\partial x_j} - R_p \frac{\partial \hat{u}_i}{\partial x_j} \right) \frac{\partial W}{\partial x_j} d\Omega \quad (3-53)$$

Using the integrals a and b of the above equation (3-53) we can rewrite, using the convex combination:

$$-1 \cdot a + R_p \cdot b = -(1 - R_p + R_p) \cdot a + R_p \cdot b = -(1 - R_p) a + R_p \underbrace{(b - a)}_{\text{Simulated second-order}}. \quad (3-54)$$

The coefficient R_p should be equal to 1, if we are to obtain the fully *second order* AV, and can be seen that setting $R_p=0$ we obtain the fully *first order* AV defined in equation (3-51). These results are going to be used for switching procedure, with the observation that constant R_p is now replaced by the calculated value of the shock sensor given by equation (3-41). It is to be noted here that in practice, as the normalized shock sensor rarely reaches zero value, the corresponding AV in case of shock will not be fully first order. However, in case of a discontinuity, we are going to use terms like “*first order dominated*” AV in the discontinuity region where the shock sensor gets a low value, and “*second order dominated*” AV in the other regions of the domain that are discontinuities free. The above first and second order AV discussion is valid when diffusion is added in streamline and gradient (cross-wind) directions.

Figure 3-6 contains an explanation of the AV switching procedure applied to obtain transonic solutions of Navier Stokes equation. The core of the algorithm is represented by the calculation, in every node of the mesh, of the shock detector value (in the flowchart as the input variable we have the pressure, but if we want to use the entropy based sensor, we need to have the density available also – see equation (3-47)). The normalisation is necessary because formula (3-41) provides a domain of variation for the sensor between order 10^{-9} and order of 10^{-2} , with the low values in regions with no shocks, and high values in regions with shocks. We need therefore a sensor value of 1 outside shocks, and close to zero value in the shock region. The sensor is then used to compute the AV term

as in equation (3-54) and determine if the situation corresponds to a first order dominated AV or a second order dominated AV. We want now to introduce other important FENSAP variables - the ratios of AV in the continuity equation to AV in the momentum and energy equations, **Figure 3-7**. The introduction of these parameters (first two fields of the “Ratio parameters” subtitle in FENSAP GUI) is explained by the type of equations we have to solve: one continuity equation, that intrinsic contains no natural viscosity terms, three scalar momentum equations, that feature natural viscosity terms in the Cauchy stress tensor, one energy equation that features as well natural viscosity.

Diffusivity, either natural or artificial, has a stabilizing effect on the system of equations. The fact that continuity equation has no intrinsic diffusivity is translated into more artificial viscosity needed for stabilization than in the momentum equation, for example. This is the reason for the use of ratios $\frac{\text{Momentum AV}}{\text{Continuity AV}}$ and $\frac{\text{Energy AV}}{\text{Continuity AV}}$. To be clearer, the ratios refer to the AV

coefficients, and not to the whole AV term, as defined by equations (2-51) – (2-55). For example, in **Figure 3-7** this AV coefficient has the value $1.0e^{-06}$. In computations it was observed that the adjustment of the ratio has more stabilizing effect than adjustment of the AV coefficient.

Going back to the diagram of **Figure 3-6**, we see that the computed value of the shock sensor is compared to a threshold, S. This value is not computed, but extracted from the fact that if the flow does not have discontinuities, sensor value has the value of around 1, i.e. second order dominated AV. For values of the sensor lower than 1, we consider the flow has first order dominated AV, **Figure 3-8**.

Implementing the switching scheme, we fixed the value of threshold at a value of 0.95. The logic of the switching procedure indicates that for values of the sensor lower than S=0.95 the ratio $\frac{\text{Momentum AV coeff.}}{\text{Continuity AV coeff.}}$ is set to $\frac{R_m}{F}$, where R_m indicates the initial ratio, and F represents a factor that divides that initial ratio to obtain

the current ratio. In this case, the flow is considered to have first order dominated AV.

For the case when the sensor value is higher than $S=0.95$, the ratio $\frac{\text{Momentum AV coeff.}}{\text{Continuity AV coeff.}}$ is set to $\frac{R_m}{G}$, where G represents a factor that divides the initial ratio R_m to obtain the current ratio. In this case, the flow is considered to have second order dominated AV. In the results section, the values for R_m , F , G and S will be given. Another interesting feature about the threshold S is that we can use it to introduce more or less first order effect into the scheme, **Figure 3-9**. The benefit of more first order in a transonic flow is that the shock capturing is better, but the disadvantage is that the first order AV introduces more oscillations in the solution. In the case of **Figure 3-9** (a) the solution will feature a first order dominated AV and a good shock capturing, while the situation in case (b) is that the second order is dominant, having the effect of a poor shock capturing. After the switching was complete for node J , the code updates $J = J+1$, and repeats the procedure. After all nodes have been used and all nodes have a value for the shock sensor and a value for AV coefficient, the normal solving procedure continues until a solution is obtained.

3.5. Mesh generation

One of the first steps in computing a numerical solution to the equations that describe a physical process is the construction of a grid. The physical domain must be covered with a mesh, so that discrete volumes or elements are identified where the conservation laws can be applied. A well constructed grid greatly improves the quality of the solution. Difficulties with numerical simulations, such as the lack of convergence to a desired level can often be explained by a poor grid quality. In this work, the meshes used are of unstructured type. In comparison with structured mesh, the unstructured meshes do not have to be segmented into blocks due to the topology of the domain and configuration of interest, a task that is time consuming for specialists working in

the field. Another advantage for an unstructured mesh, e.g. a 3D tetrahedrons mesh, is that the flow behaviour is better captured in all three space directions.

The commercial software ICEM CFD was used as a tool to generate the initial grids for the test cases presented in the following chapters. Because the problem in discussion includes viscous effects, the mesh sought has to have a high density in regions with viscous interfaces, which in our case lie around the solid wall of the wing or airfoil. For a better resolution of the boundary layer, a mixed mesh is used: close to the wall a number of prism layers are placed, while the rest of the domain is filled with tetras. A hybrid mesh is obtained in this manner, with the observation that in the generation process the elements with a pyramid shape should be avoided, as they are not accepted either by the flow solver or the mesh adaptation software.

In Chapter 4 a 2D flow solution is obtained for RAE 2822 airfoil. The mesh used in this simulation is three dimensional with symmetric boundary conditions (BC) at the root and tip. The inputs that ICEM CFD requires in order to generate a mesh are the topology, in the form of *.tin file, and the boundary conditions generically designated as family_boco file.

3.6. Mesh adaptation

During a typical CFD analysis, an important effort is devoted to the mesh generation process. Mesh generation is not only time consuming, but also the resulting mesh may not be completely appropriate to sufficiently resolve all details of the flow. Mesh adaptation methodology is a powerful tool, not only to improve solver accuracy but also to allow simulations to start from an arbitrarily coarse initial grid. OPTIGRID, a robust and efficient 3D automatic mesh adaptation with CAD integrity tool, was used in the present work. The adaptation procedure uses an *a posteriori* interpolation error estimate, whose magnitude and direction are controlled by the matrix of local second derivatives of a selected flow variable. This error is projected over the mesh edges, and drives the nodal movement algorithm, as well as the edge refinement, coarsening and face swapping strategies, [17]. The adaptation process, however, can take

several solution – adaptation cycles, depending on the quality of the initial grid and the complexity of the test case. Chapter 5 contains detailed description of how OPTIGRID cycles have been used for the ONERA M6 wing test case. OPTIGRID offers automatic solution-based anisotropic mesh adaptation, as well as geometry based mesh smoothing of the initial mesh, even before a solution is attempted. The latter feature is quite important, as it is not unusual for a CFD code to fail simply because of a poor initial grid, e.g., grids with cells that have negative volumes, are too skewed or are degenerate. Once a solution is launched, the basic adaptation operation includes node movement, edge refinement, coarsening and swapping for hybrid grids consisting of any combination of tetrahedrons, prisms, hexahedra and pyramids.

The solution-based adaptation is driven by an *a posteriori* error estimation based on the Hessian of a selected scalar flow variable, since for FEM-FVM solvers with linear basis functions the truncation error is dominated by the second derivatives. The eigenvectors and the eigenvalues of H give the local direction and the local magnitude of the stretching, respectively. In this manner, anisotropy is gradually created. The goal of the adaptation is to equally-distribute the error on the adapted grid, where the error along an edge in the Riemannian metric is computed as:

$$\varepsilon = \int_0^1 \sqrt{\mathbf{x}^T H(s) \mathbf{x}} ds \quad (3-55)$$

where $\mathbf{x}$ is the vector that defines the edge, and H is the absolute value of the Hessian matrix of the adaptation variable,[17].

The sequence of operations begins with node movement, edge refinement and edge swapping on solid boundaries, to satisfy a minimum and maximum edge length constraint, as well as a curvature constraint, yielding substantial surface CAD improvements. The process then continues with node movement in the entire domain, followed by refinement and coarsening, then swapping, before concluding with additional node movement. OPTIGRID preserves CAD integrity by re-projecting boundary points onto the original surfaces during the adaptation process.

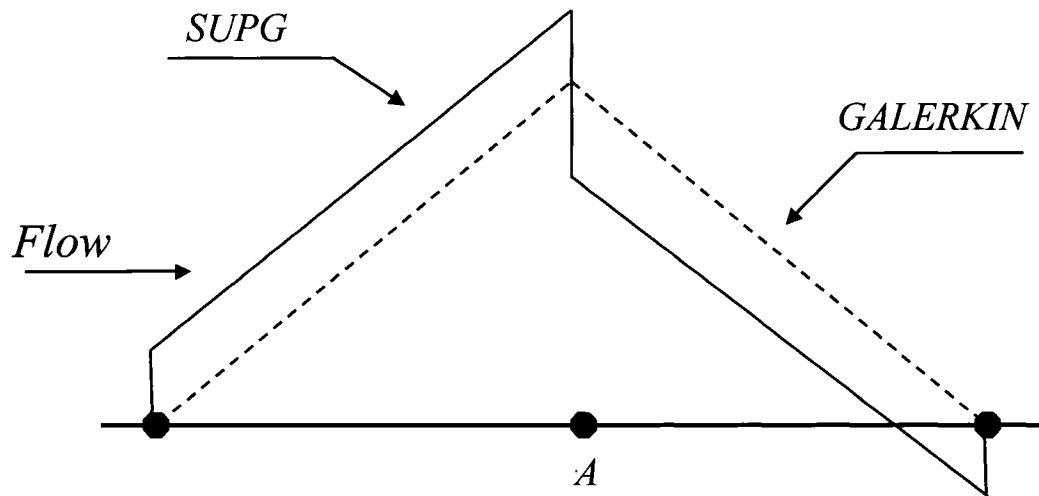


Figure 3-1: Comparison between linear Galerkin and Petrov-Galerkin weights

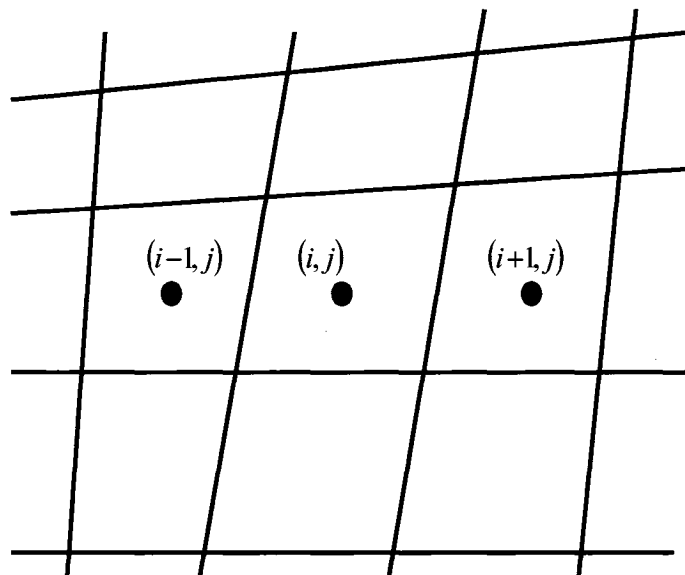


Figure 3-2: Arbitrary 2D mesh used to illustrate the discontinuity detector definition for a finite volume scheme

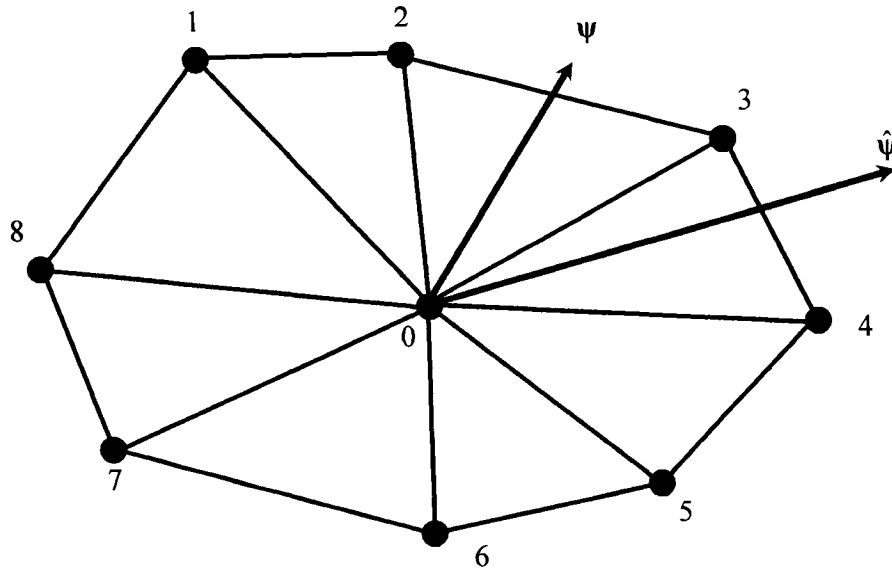


Figure 3-3: Arbitrary unstructured 2D mesh used to illustrate shock detector, local and global derivative definitions

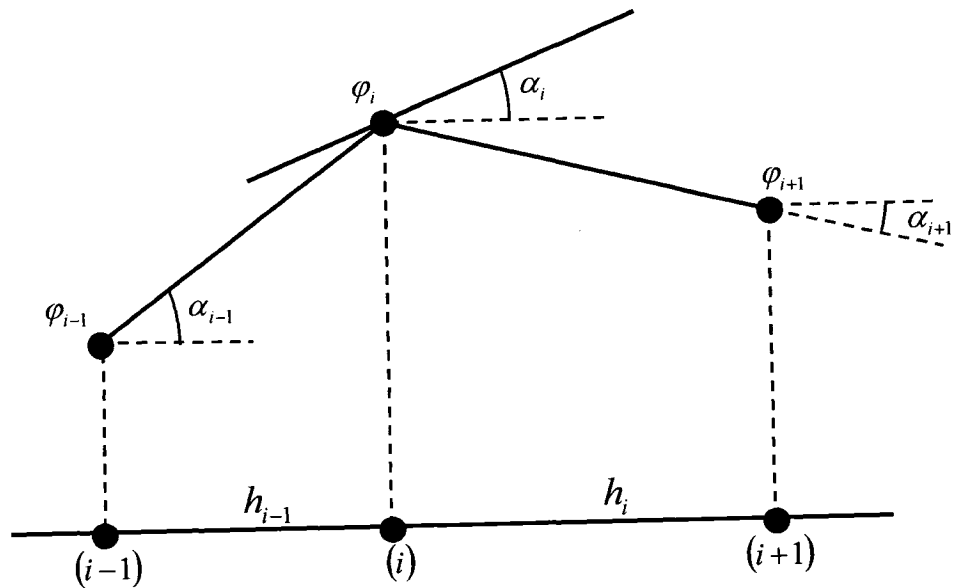


Figure 3-4: Global and local derivatives for 1D flow

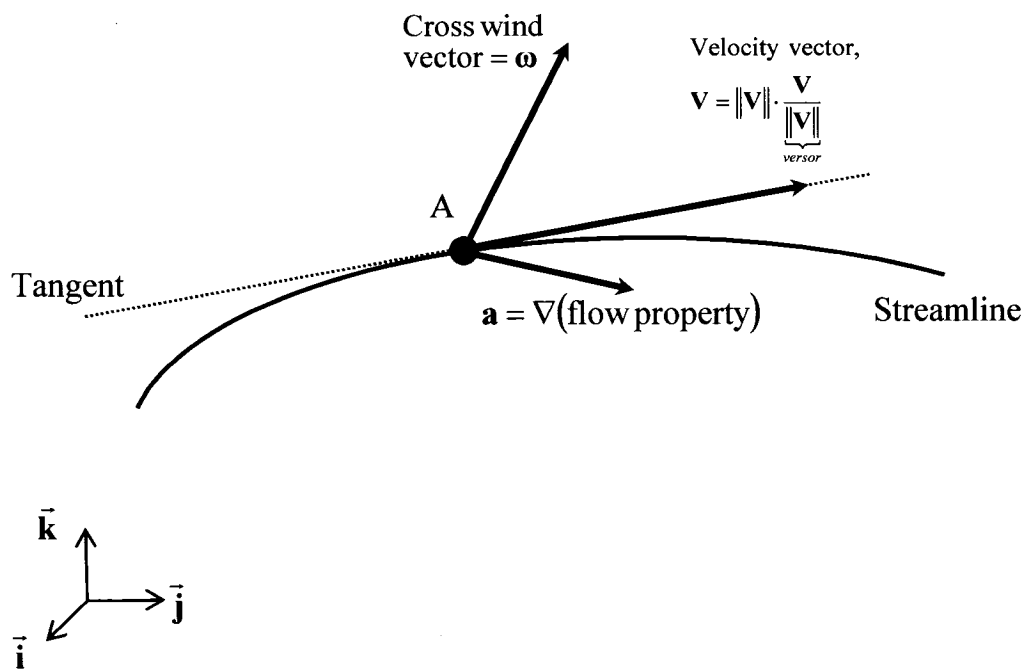


Figure 3-5: Streamline and cross wind directions at mesh point A

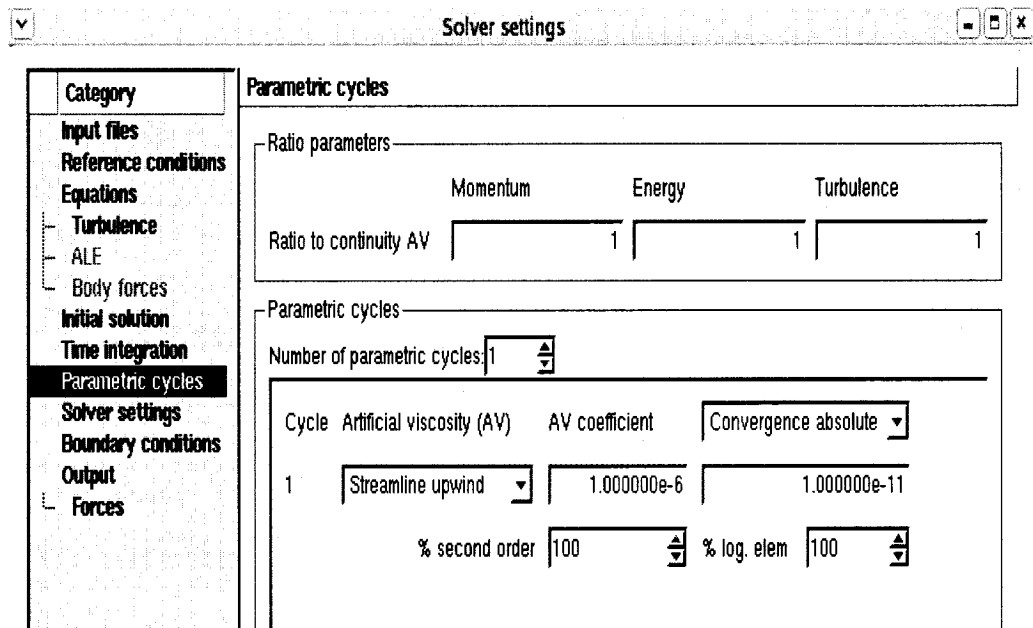


Figure 3-6: Artificial Viscosity ratios as defined in FENSAP Graphical User Interface (from NTI – fensapiceGUI executable)

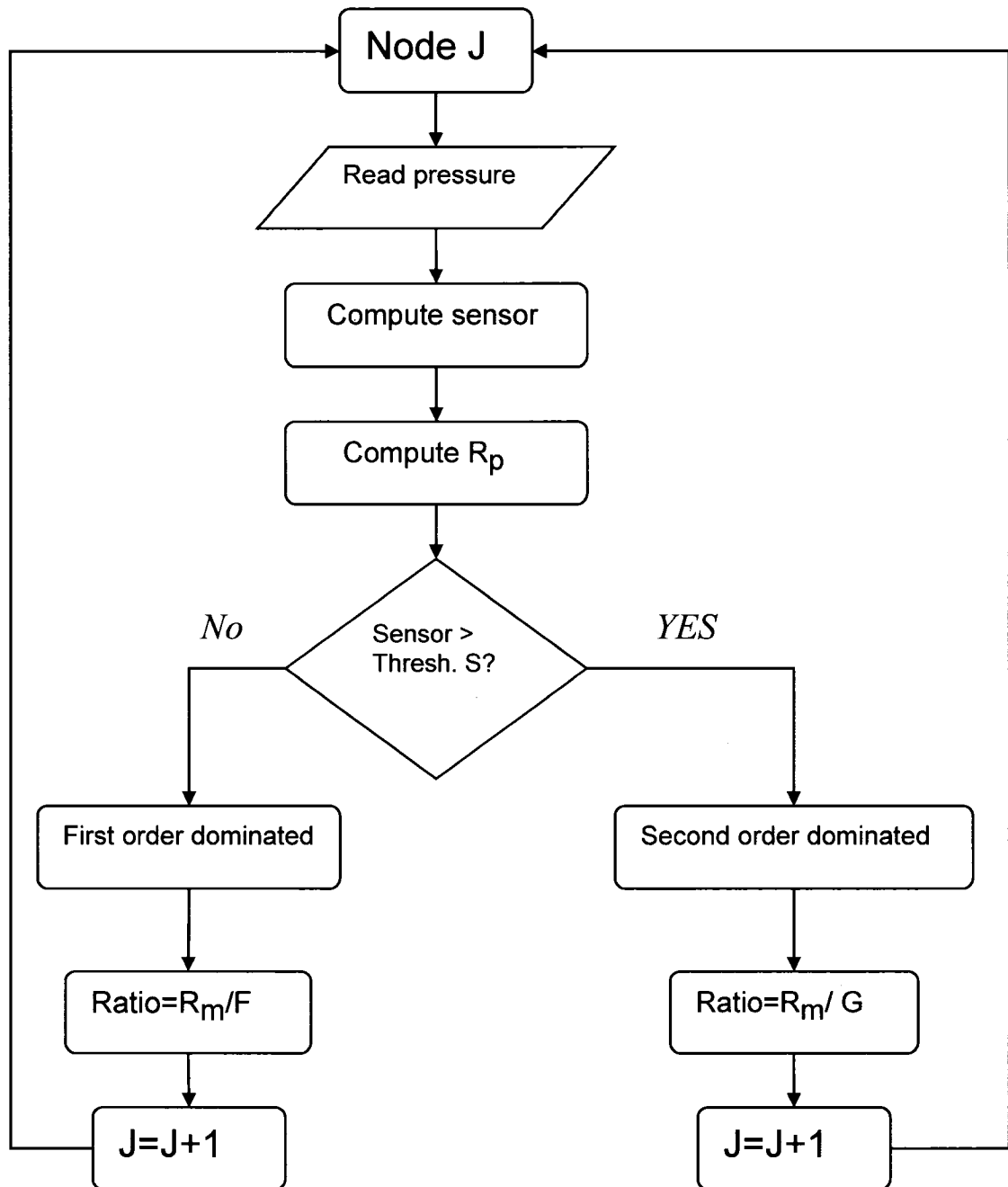


Figure 3-7: Switching procedure

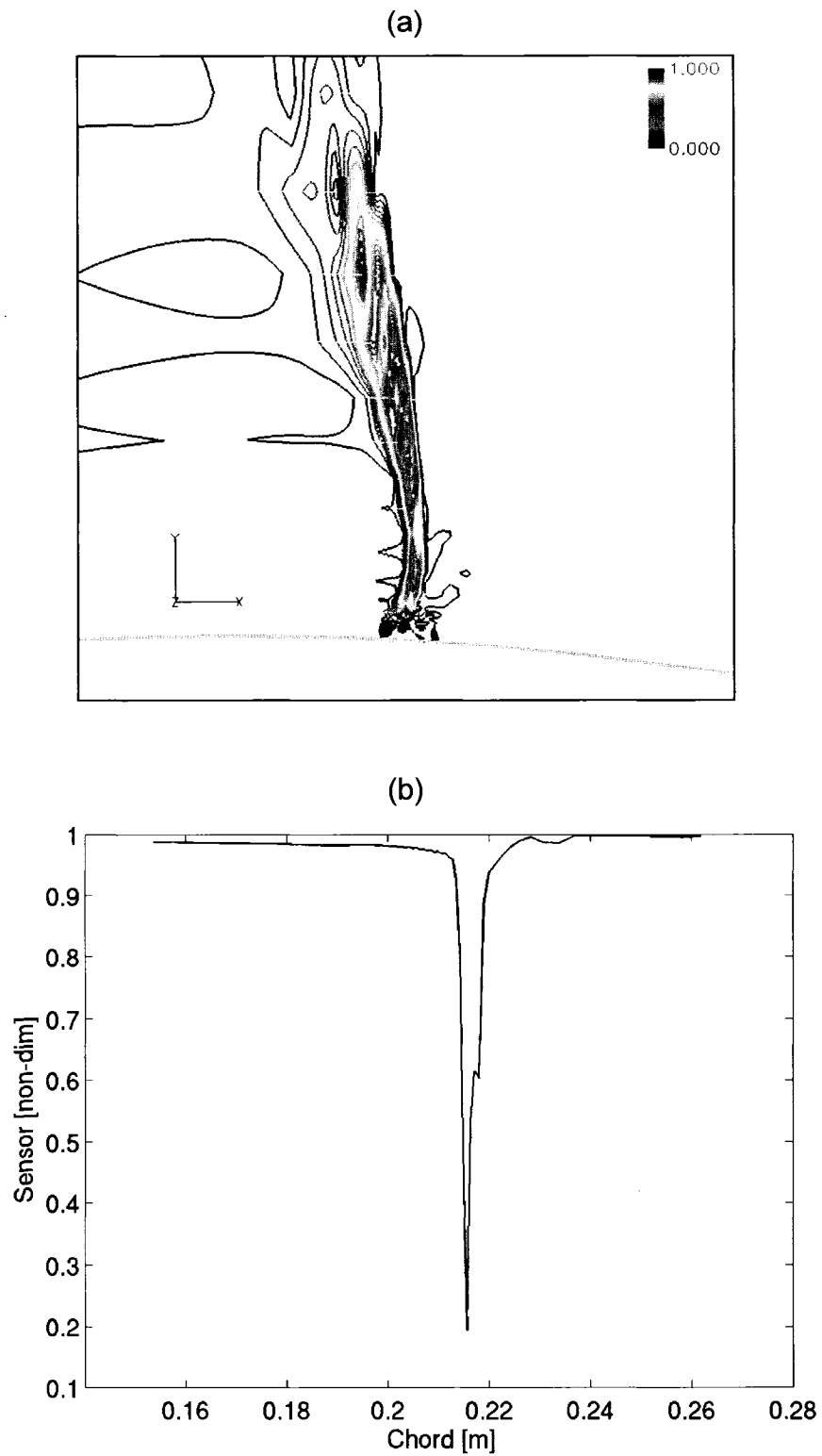


Figure 3-8: (a) Shock sensor contours for a 2D transonic viscous flow around RAE 2822 airfoil. (b) Shock sensor variation through the shock wave

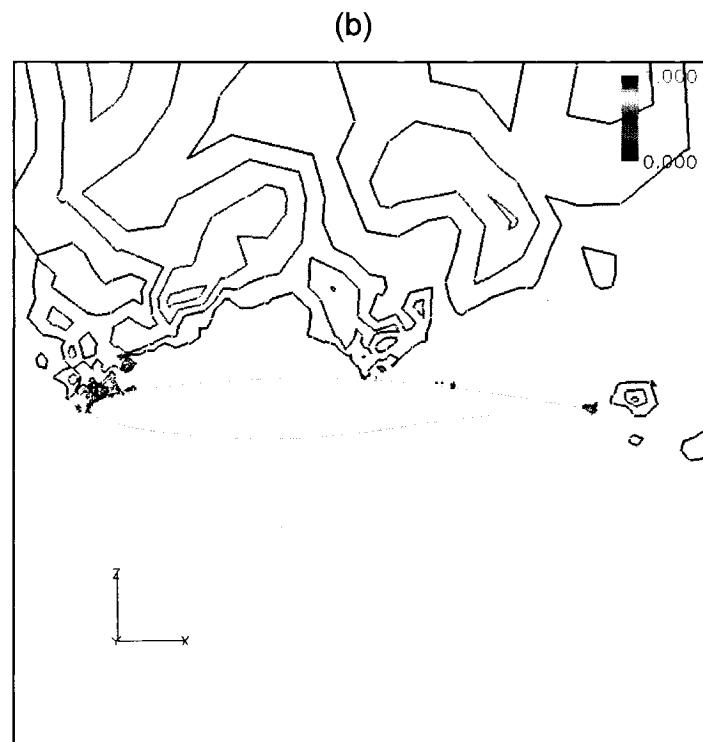
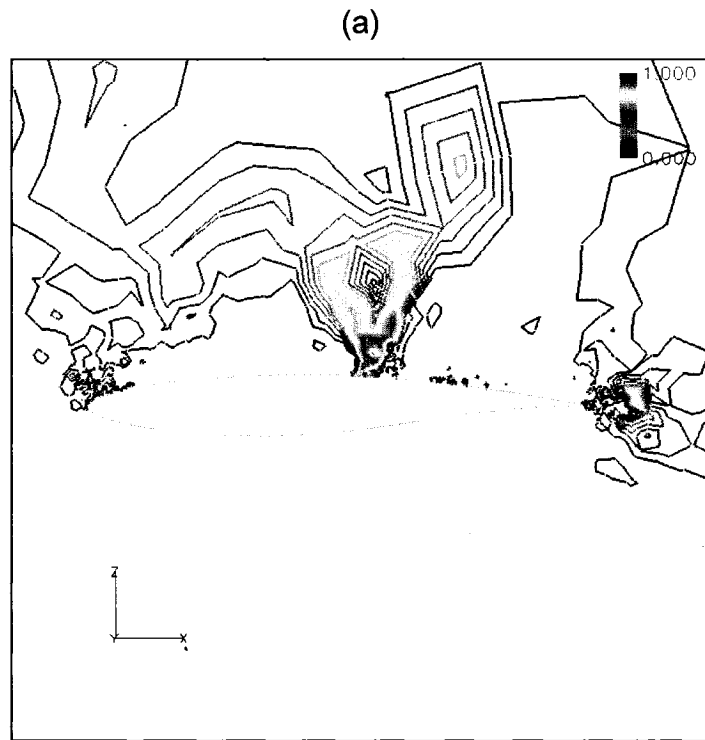


Figure 3-9: RAE 2822 transonic viscous flow: (a) Sensor contours for threshold $S=0.99$ –first order dominated AV coefficient (b) Sensor contours for threshold $S=0.9$ –second order dominated AV coefficient

4. Numerical results - RAE 2822 airfoil test case

To validate the implementation of the shock detector within the SU/SUPG scheme, two standard test cases have been chosen: the transonic airfoil RAE 2822, which will be shown in this chapter, and the ONERA M6 wing test case that constitutes the subject of the next chapter.

The main tool used in this analysis was FENSAP, as the flow solver. ICEM CFD commercial software was used for mesh generation and Field View commercial software was used for solution visualization. The simulations were performed at McGill's CFD LAB. Computers used were CFD LAB's 16 CPU SGI ORIGIN cluster, CLUMEQ's SGI Origin 3800 cluster (64 CPUs, 128Mb RAM, 600MHz) and CLUMEQ's Beowulf Cluster (256 CPUs, 1.5Gb RAM , 1.6 GHz).

4.1. Geometry

The computed RAE2822 test case corresponds to the experimental condition 9 listed in [16]. The dimensions used to define the airfoil for the present solution are given in the **Table 4-1** below:

Characteristic length	0.3809 m
Reference area	0.0171 m ²
Span	0.045 m

Table 4-1: RAE 2822 reference dimensions

The computational domain is of C type, with 15 chord lengths spacing in front of the leading edge, after the trailing edge, above the airfoil and below the airfoil. The airfoil detailed geometry is given in Appendix A.

4.2. Flow parameters

The performed simulations consider the RAE 2822 airfoil at low angle of attack (AOA), low transonic Mach number and high Reynolds number, as follows:

Mach number (M)	0.73
Reynolds number (Re)	6.5 million
Angle of Attack (AOA)	2.79 degrees
Free stream static pressure	101325 Pa
Free stream static temperature	288 K

Table 4-2: Reference conditions for RAE 2822 transonic simulations.

In this work, the one equation Spalart-Allmaras (SA) turbulence model was used to obtain transonic viscous simulations. To investigate if the transition length can be neglected, we performed a simulation considering 10% of the chord as the transition length, and then compared this result with the result where flow was considered turbulent starting right from the leading edge. The error between the two cases was within 10^{-03} %, therefore the flow is considered to be turbulent from the very beginning and the transition length is neglected for the following simulations.

The other important settings are the AV coefficient, CFL number and the GMRES parameters. These parameters are used to control the quality of the solution (AV) and the code convergence (CFL, GMRES). For each simulation the values of the above parameters will be indicated.

Through numerical experiments at the CFD Lab it has been determined that FENSAP, for the 2D test case RAE 2822 airfoil, requires approximately 15 chord lengths between the wall and the location of the external boundary.

4.3. Computation mesh

The grid used in this chapter for 2D flow computations is in fact 3D. The wall is represented by the rectangular wing described in [16], and the free stream velocity vectors are situated in planes normal to the wing's longitudinal axes. The grid is unstructured and the cells are in the shape of tetrahedrons and prisms. A large number of cells are placed in the vicinity of the solid boundary, in order to resolve the boundary layer.

The complete domain of the mesh and the leading edge and trailing edge details are shown in **Figures 4-1** and **4-2**. The total number of elements contained in

the grid is 282,595 tetrahedrons and 160,710 prisms organized in 15 layers. The total number of nodes contained in the domain is 141,756, the approximate number of nodes on the wall is 5,673 and the file size in ASCII format is 21Mb. Another important characteristic of this grid is the minimum grid spacing near the wall. A value of approximately 3e-06 m was chosen to ensure the proper capture of the boundary layer.

4.4. Discussion of results

The present solution accuracy is measured through comparisons to wind tunnel measurements and other CFD solutions. The relative error between aerodynamic coefficients is computed based on a simple formulation shown in the equation (4-1) below.

$$\varepsilon_{C_l} = \frac{|C_{l\text{ computation}} - C_{l\text{ test}}|}{C_{l\text{ test}}}, \quad \varepsilon_{C_d} = \frac{|C_{d\text{ computation}} - C_{d\text{ test}}|}{C_{d\text{ test}}} \quad (4-1)$$

Other parameters, like the pressure distribution on the airfoil, are compared graphically superimposing experimental data, other CFD data and present solution.

In this section an unstructured mesh was used to obtain the solution of the system of equations. Three separate cases are investigated and each case is based on the same numerical scheme and flight conditions. However, they differ on the type of artificial viscosity method used. The three approaches are:

- Streamline Upwind (SU) with fully first order AV option
- SU with fully second order AV option
- SU plus the shock detector implementation

When the fully first order AV is selected, the coefficient R_p in equation (3-54) is set to 0, which corresponds to a value of 0 in the field “%second order” of **Figure 3-6**. In this case we expect a solution that shows oscillations, particularly in regions of high gradients. When the fully second order AV is selected, the coefficient R_p in equation (3-54) is set to 1, which corresponds to 100 in the field “%second order” of **Figure 3-6**. In this case the simulated second order AV is

used, and the solution is expected to show fewer oscillations than in the fully first order case. The third case refers to the blended first and second order AV scheme based on the shock detector, and is expected to provide better results than the previous two cases.

4.4.1. SU scheme without shock detector

The SU scheme as it is implemented in FENSAP and as we discussed in Chapter 3, has the capability of choosing the AV order from the beginning. The limits we discuss here in terms of AV order are fully first order AV and fully second order AV, but any combination of first and second order AV can be selected. An important parameter used to obtain the solution is the ratio R_m as defined in Chapter 3; it has different values for first and second order AV, as will be indicated below. By simulating separately fully first and second order AV, we want to determine the limitations introduced in each case, and to prepare the implementation of a scheme that automatically exploits the advantages that each AV order has to offer.

4.4.1.1. Fully first order AV scheme

The fully first order AV scheme is expected to provide sharp shock capturing with overshoots and oscillations in the solution before and after the shock wave. To obtain the final solution for this case, we used the restart procedure, i.e. the solution obtained in a parametric cycle becomes the initial solution for the next parametric cycle. All settings are unaltered from one parametric cycle to the next, with the exception of AV and/or R_m . Hence, the first simulation A has as initial conditions the boundary condition values of flow variables for the first parametric cycle, and $AV=0.1$ (see **Table 4-3**). The second parametric cycle is restarted from the first one, to produce a second intermediate solution that will be the initial condition for the third parametric cycle. The process continues in this manner, and simulation A ends with the tenth parametric cycle that has $AV=5e-5$ and $R_m=1.0$. The obtained solution shows a good convergence, but is

very diffusive and shows no shock capturing. From this solution, we restart simulation B, continuing to reduce the diffusivity in the system from one parametric cycle to the next by reducing the AV coefficient, but maintaining $R_m=1.0$ (see **Table 4-3**).

The final solution of simulation B is still very diffusive, but from this solution we restart another simulation with $AV=1e-09$ and $R_m=0.5$. In this way, modifying ratio R_m we reduce the diffusivity in the momentum equation and improve the shock capturing. The solution does not show a shock wave, but the trend is good. After another restart procedure with $AV=1e-09$ and $R_m=0.2$, we obtain a less diffused solution and the formation of a weak shock. We performed then a last restart and reduced further the diffusivity in the momentum equation, to obtain the final solution for this case. The solution shown in **Figure 4-3** and **Figure 4-4** was obtained with an $AV = 1e-09$, $R_m = 0.08$, $CFL = 500$ and default settings for GMRES.

Parametric cycle #	AV coefficients for simulation A	AV coefficients for simulation B
1	0.1	1e-05
2	0.07	7e-06
3	0.03	4e-06
4	0.009	1e-06
5	0.006	7e-07
6	0.002	3e-07
7	0.0008	9e-08
8	0.0004	5e-08
9	9e-05	1e-08
10	5e-05	5e-09

Table 4-3: AV coefficients corresponding to each parameter cycle of simulations A and B

A sharp shock capturing was obtained only after a decrease of the ratio R_m to a value of 0.08. For a higher diffusivity of the momentum equation (i.e. ratio R_m higher than 0.08) the shock was not captured or was smeared and for a lower value (i.e. ratio R_m lower than 0.08) there are large oscillations before and after the shock. For the final solution, the overall matching against test data is off and

the oscillations are present and more visible towards the trailing edge (TE) , but the captured shock is sharp and a cross section through the shock wave shows a Mach number variation without non-physical overshoot-undershoot. We retained the positive aspect of this scheme (the shock capturing) for a future implementation of an automatic computational device that sets the proper AV in the system of equations based on the presence of flow property discontinuities. **Figure 4-3** and **Figure 4-4** show the pressure distribution and Mach number variation for the final solution of this case.

4.4.1.2. Fully second order AV scheme

For this case, the solution is expected to show a good shock capturing and fewer oscillations, as a result of the second order AV smoothing effect. The restart procedure explained in the previous subsection was also used to obtain the final solution for this case. The first set of 10 parametric cycles form simulation A, with the AV coefficients shown in **Table 4-3** and ratio $R_m=1.0$. The obtained solution is very diffusive, in spite of the good convergence (**Figure 4-5 (a)**), and we therefore proceed to simulation B, which is a restart from the final solution of simulation A. In simulation B we further progressively reduce the AV coefficient (**Table 4-3**) but keep $R_m=1.0$. The final solution of simulation B (convergence shown in **Figure 4-5 (b)**) is still diffusive, and the shock is not properly captured.

The overall residual level increases a little as we decrease the artificial viscosity coefficient. The recommended value for this coefficient is around $1E-6$ and for this value we obtained a convergence level of order $1E-11$. The residual level increases only after we decrease the artificial viscosity coefficient below the $1E-6$ limit, in our attempt to capture a clean shock. The artificial viscosity added to the system helps stabilize the scheme and obtaining a low convergence level. A compromise between convergence level and shock capturing was reached at an artificial coefficient of approximately $1E-10$.

The level of convergence for the lift and drag coefficients of simulations A and B is given in **Figure 4-6**, **Table 4-4** and **Table 4-5**. The difference between the

current value of drag and the value from the previous iteration is plotted in logarithmic scale as shown in **Figure 4-6 (b)**, and a similar diagram is shown for the lift coefficient in **Figure 4-6 (a)**.

	Iteration 2000	Iteration 2500	Iteration 3000
Cl	0.800117	0.808272	0.809462
Cd	0.021915	0.022466	0.0225122

Table 4-4: Aerodynamic coefficients for the iterations 2000, 2500 and 3000

	From 2000 to 2500	From 2500 to 3000
Relative error Cl	1.01%	0.14%
Relative error Cd	2.5%	0.21%

Table 4-5: Relative error for aerodynamic coefficients

The best solution was obtained for $R_m=0.5$ and the Mach contour lines, pressure distribution, Mach variation and convergence are given in **Figure 4-7** to **Figure 4-9**. The contour lines shown are smoother and the pressure distribution matching is more accurate than in the previously discussed fully first order case. The CFL was set to 500, $AV=1e-10$ and the default setting for GMRES. The convergence after 300 iterations dropped three orders of magnitude with an overall residual of approximate 10^{-09} and the Spalart-Allmaras residual decreased two orders of magnitude and stabilized at approximate 10^{-05} . The position of the shock is obtained closed to the middle of the airfoil, as the test indicates, but the peak values in the pressure distribution diagram are not reached, generating a lower coefficient of lift than in the experiment. The Mach number variation through the shock wave displays an overshoot that is not physical, this overshoot being observed in other computations that used second order AV schemes. The introduction of the shock wave detector will improve this overshoot behavior, switching off the second order AV terms in the vicinity of the discontinuities.

4.4.2. SU scheme with shock detector

Knowing from the previous two subsections about the advantages and disadvantages that either first order or second order AV schemes introduce, we developed an AV scheme that for transonic viscous flows automatically diminishes the contribution of the second order AV terms in the cells that have been identified of having a value of the corresponding shock sensor lower than a predefined threshold (parameter S). This implementation introduced in the flow solver code three new parameters S, F and G, and their influence was discussed in Chapter 3. The final solution of this case was obtained through the restart procedure previously discussed in the fully first order AV subsection. In the final restart cycle, an initial setting of $S=0.99$, $F=10$ and $G=1$ was tried generating a first order AV dominated solution with oscillations in the region of the shock. To improve the situation, the contribution of each parameter S, F and G has been analyzed leading to the conclusion that parameter F has a dominant effect against the other two parameters S and G. We assigned for the parameters S and G the values $S=0.95$ and $G=1.4$ and numerical experiments were performed in order to determine the most appropriate value for the parameter F.

The best solution was obtain for $F=8$ while the other parameters of the flow solver were $AV=1e-10$, $CFL=100$, GMRES set for 510 iterations for the momentum equation and 160 for the turbulence equation (with the remark that this increased number of iterations improves the solution accuracy, but slows down the computation). On a 16 CPU execution, one time step was performed in about 38 seconds, with the total time for this simulation of 3.27 hours. The final solution of this case (see **Figure 4-10** to **Figure 4-13**) is in good agreement with the experimental data and other CFD results. The Mach contour lines indicate a smooth solution, and a good shock capturing. The pressure distribution matches the test data, and the overshoot in the shock region is attenuated.

The final parametric cycle has 300 iterations, but after the first 100 iterations the overall residual drops one order of magnitude and stabilizes at approximately 10^{-08} while the coefficients of lift and drag reached steady state.

Since this is a restart solution, the overall residual drops globally from 10^{-01} to 10^{-08} . The aerodynamic coefficients are shown in **Table 4-6** and **Table 4-7**.

	Experimental results,[34]	Computation – present solution	Relative error ε [%] - eq.(4-1)
Lift Coefficient C_l	0.803	0.8218	2.34
Drag Coefficient C_d	0.0168	0.01963	14.4

Table 4-6: C_l and C_d from experiment and present solution (hybrid mesh and SU + shock detector)

	Other CFD,[34]	Computation – present solution	Relative error ε [%] - eq.(4-1)
Lift Coefficient C_l	0.8415	0.8218	2.34
Drag Coefficient C_d	0.0181	0.01963	8.45

Table 4-7: C_l and C_d for other CFD and for present solution (hybrid mesh and SU + shock detector)

The aerodynamic coefficients for 1st and 2nd order artificial viscosity are shown in the **Table 4-8** below:

	Experimental results,[34]	Other CFD,[34]	Computation – SU + shock detector	Computation – 1 st order AV	Computation – 2 nd order AV
C_l	0.803	0.8415	0.8218	0.7058	0.8141
C_d	0.0168	0.0181	0.01963	0.02449	0.02038

Table 4-8: C_l and C_d comparison

4.5. Comparison between SU and SU with shock detector

To evaluate the performance of the newly introduced AV scheme within SU we superpose the pressure distribution of the cases discussed in the previous three subsections (see **Figure 4-14**).

Both fully second order and shock sensor solution are in good agreement with experimental data, while the fully first order shows a disagreement for the shock position and peak pressures. The positive aspect of fully first order AV solution is the sharp shock capturing and this feature was exploited by the SU + sensor algorithm. The shock sensor solution shows almost no overshoot in the shock region, and the suction portion of this solution matches better to experimental data. The automatic computation of the shock sensor for any mesh point (parameter R_p of equations (3-53) and (3-54)) and the dynamic allocation of the AV in the computational domain based on the presence of discontinuities made a significant improvement to the solution accuracy for the transonic viscous flows around the RAE 2822 airfoil.

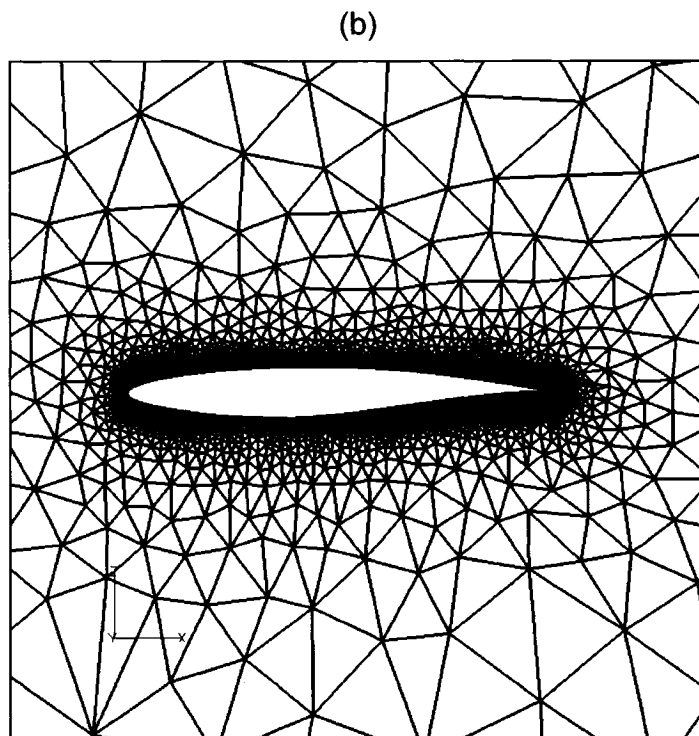
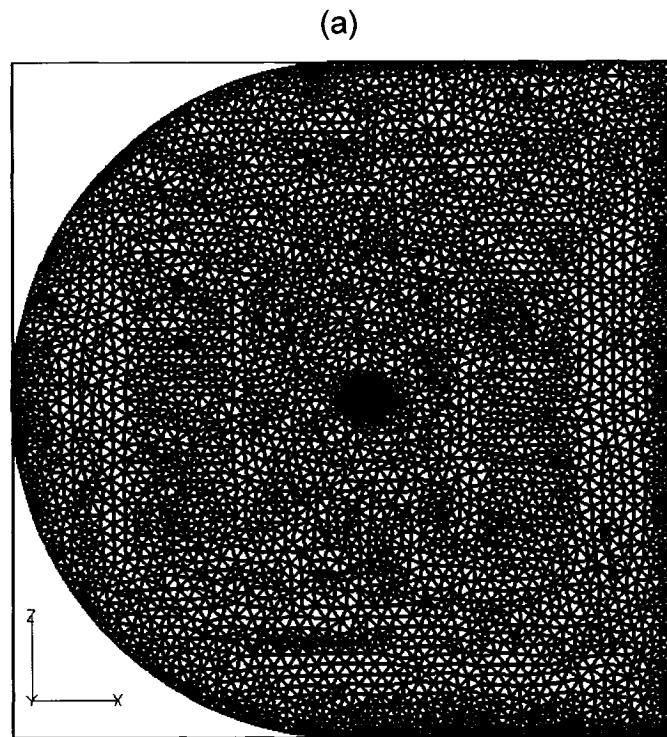
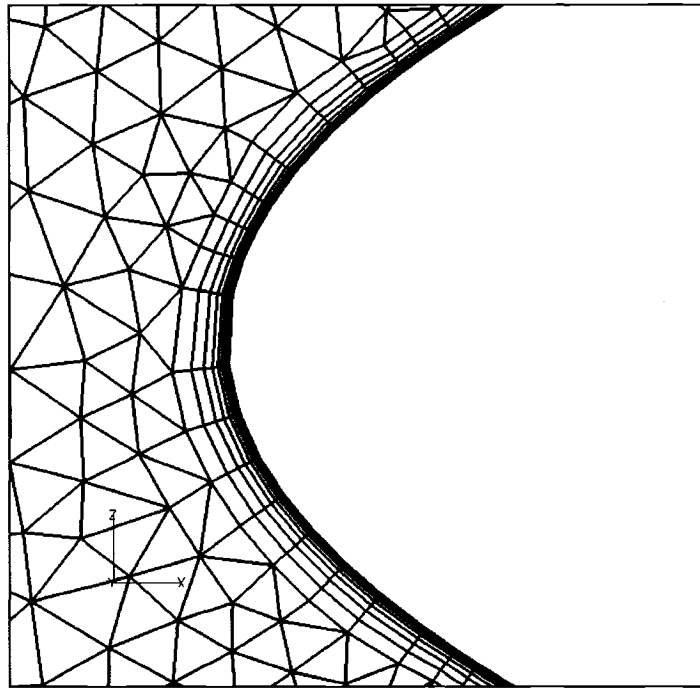


Figure 4-1: (a) RAE 2822 Hybrid Mesh (tetras and prisms). (b) Airfoil detail.

(a)



(b)

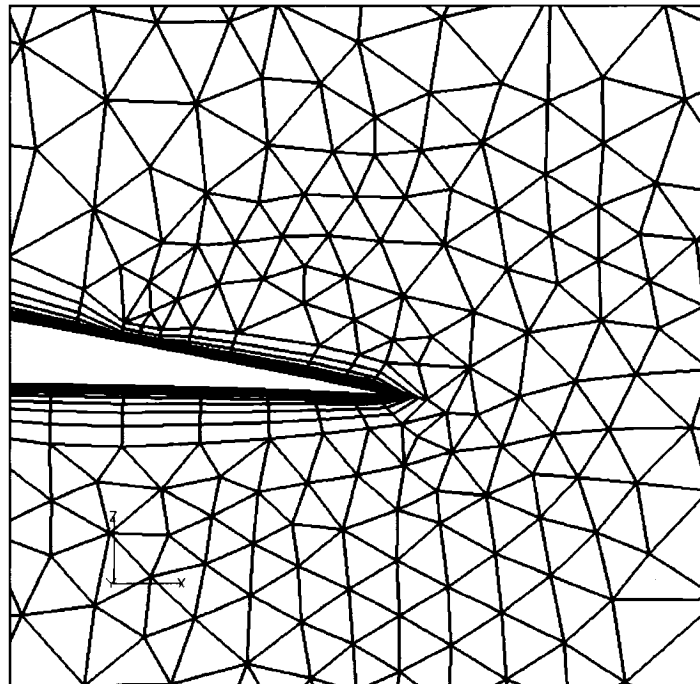


Figure 4-2: (a) RAE 2822 Leading edge detail. (b) RAE 2822 Trailing edge detail.

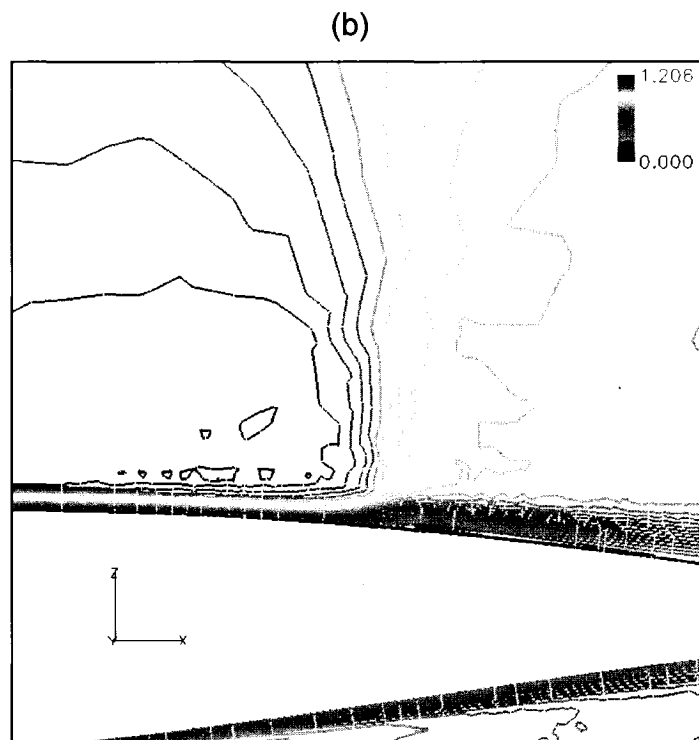
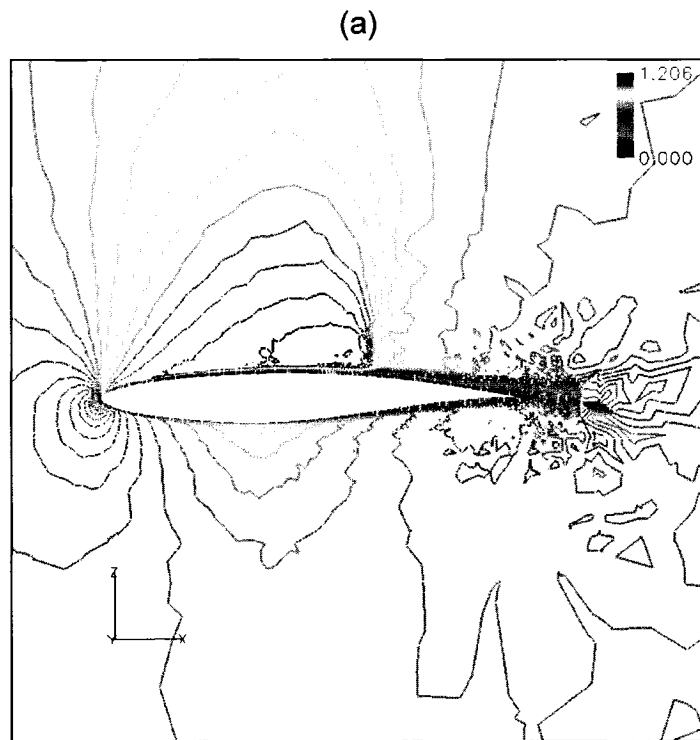


Figure 4-3: RAE 2822 transonic viscous solution using tetra mesh and SU fully first order AV scheme. (a) Mach contours. (b) Shock wave detail.

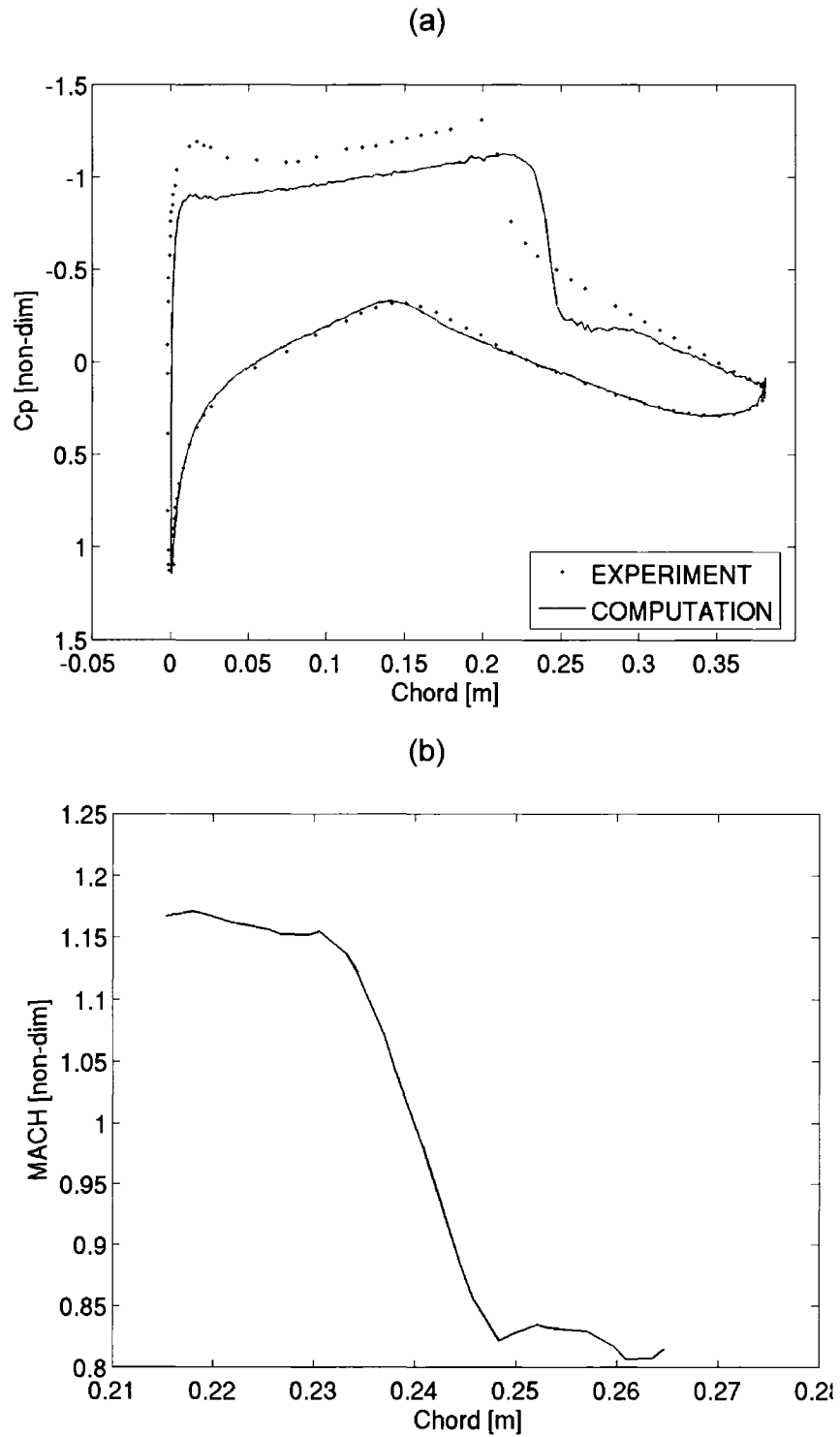


Figure 4-4: RAE 2822 transonic viscous solution using tetra mesh and SU fully first order AV scheme. (a) Wall Coefficient of Pressure distribution – experimental and present solution data. (b) Mach number variation through the shock wave.

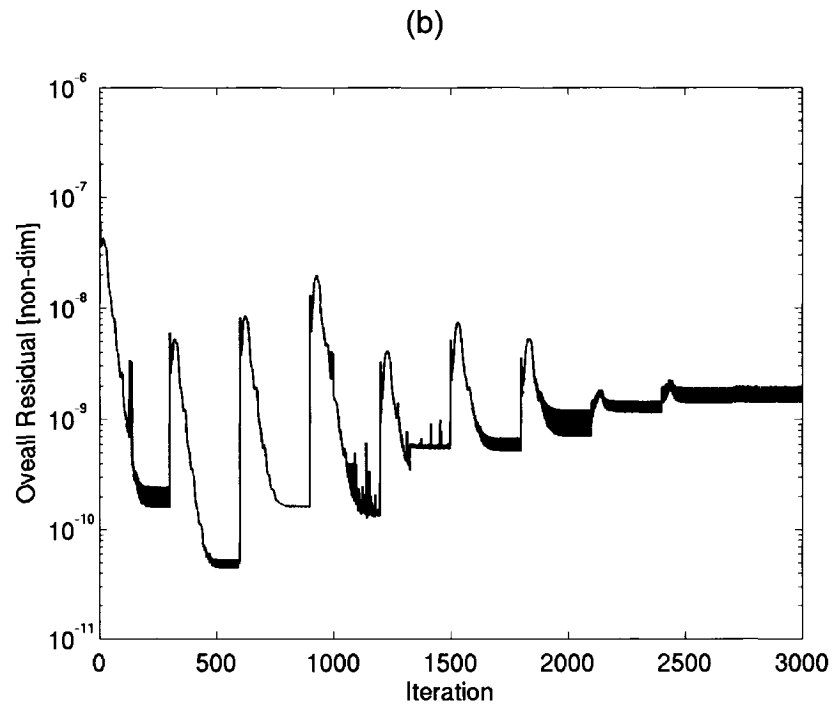
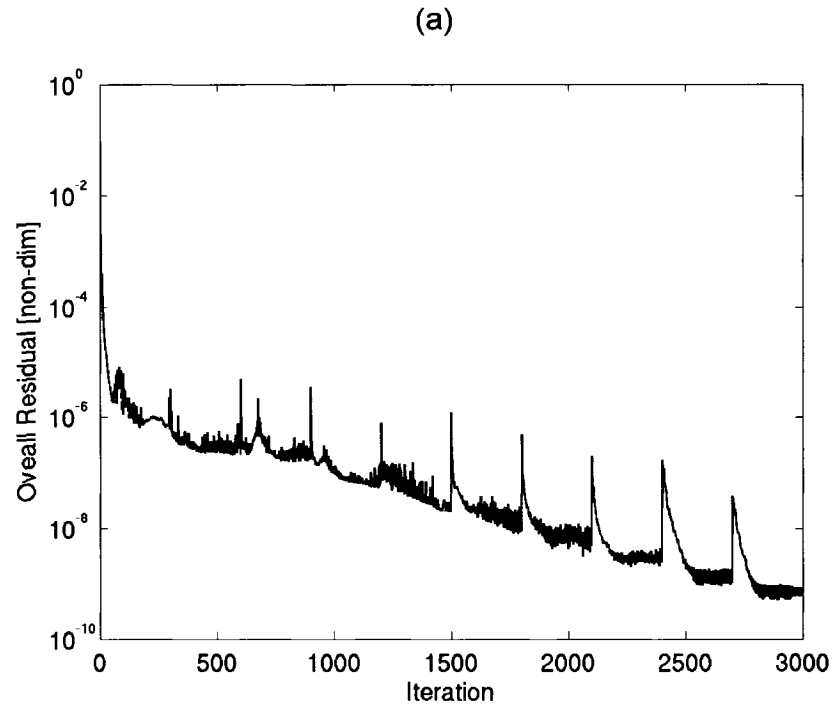


Figure 4-5: Overall Residuals for fully second order AV case: (a) Simulation A.
(b) Simulation B.

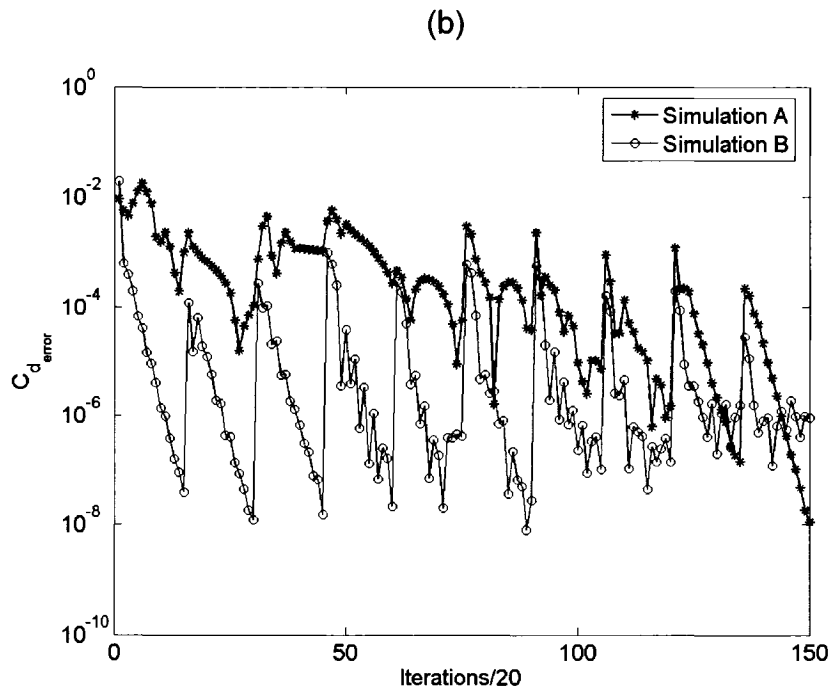
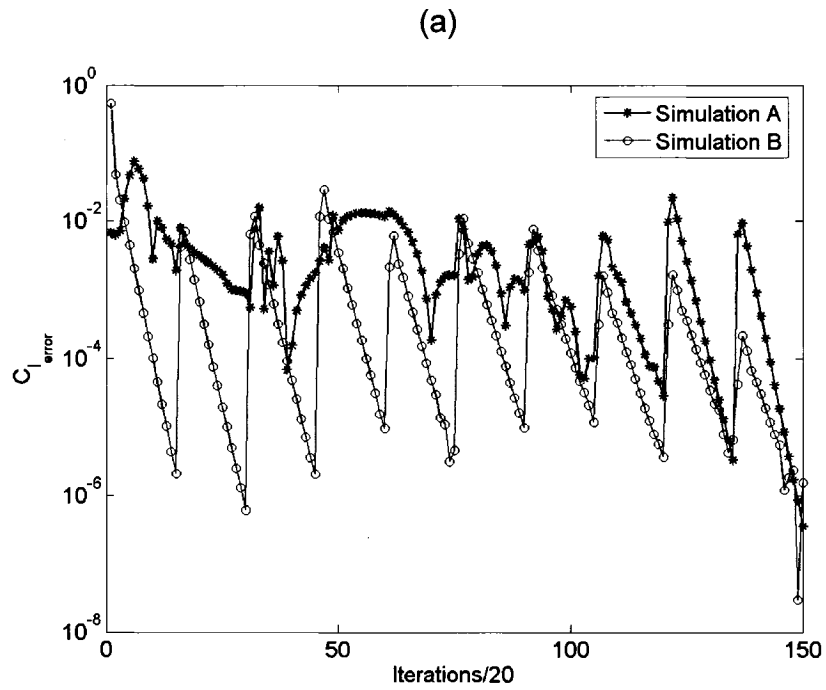


Figure 4-6: C_l and C_d convergence for fully second order case: (a) Coefficient of lift convergence for the restart solutions A and B. (b) Coefficient of drag convergence for the restart solution A and B.

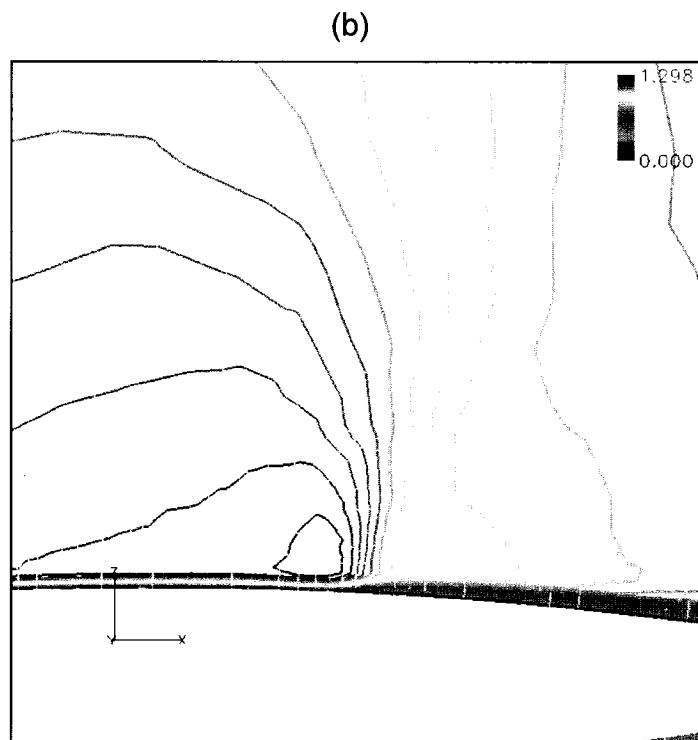
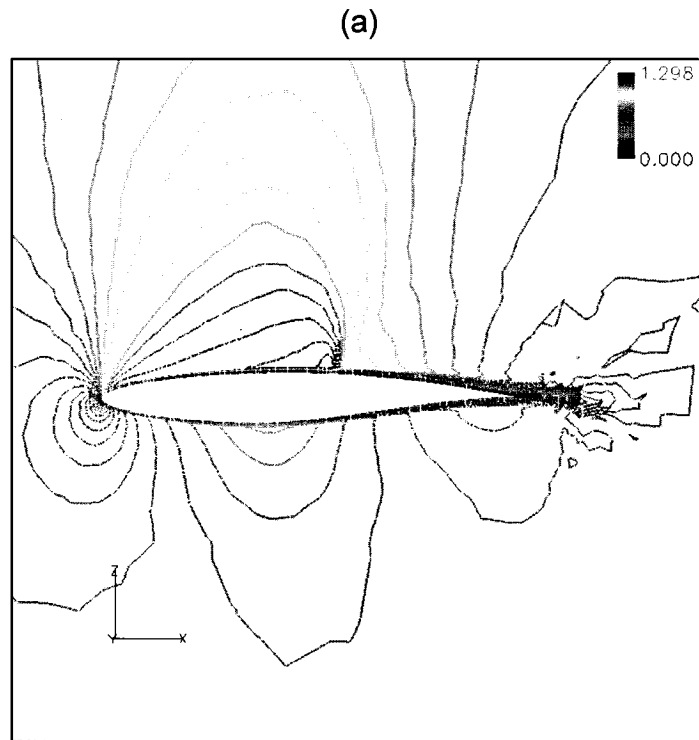


Figure 4-7: RAE 2822 transonic viscous solution using tetra mesh and SU fully second order AV scheme. (a) Mach contours. (b) Shock wave detail.

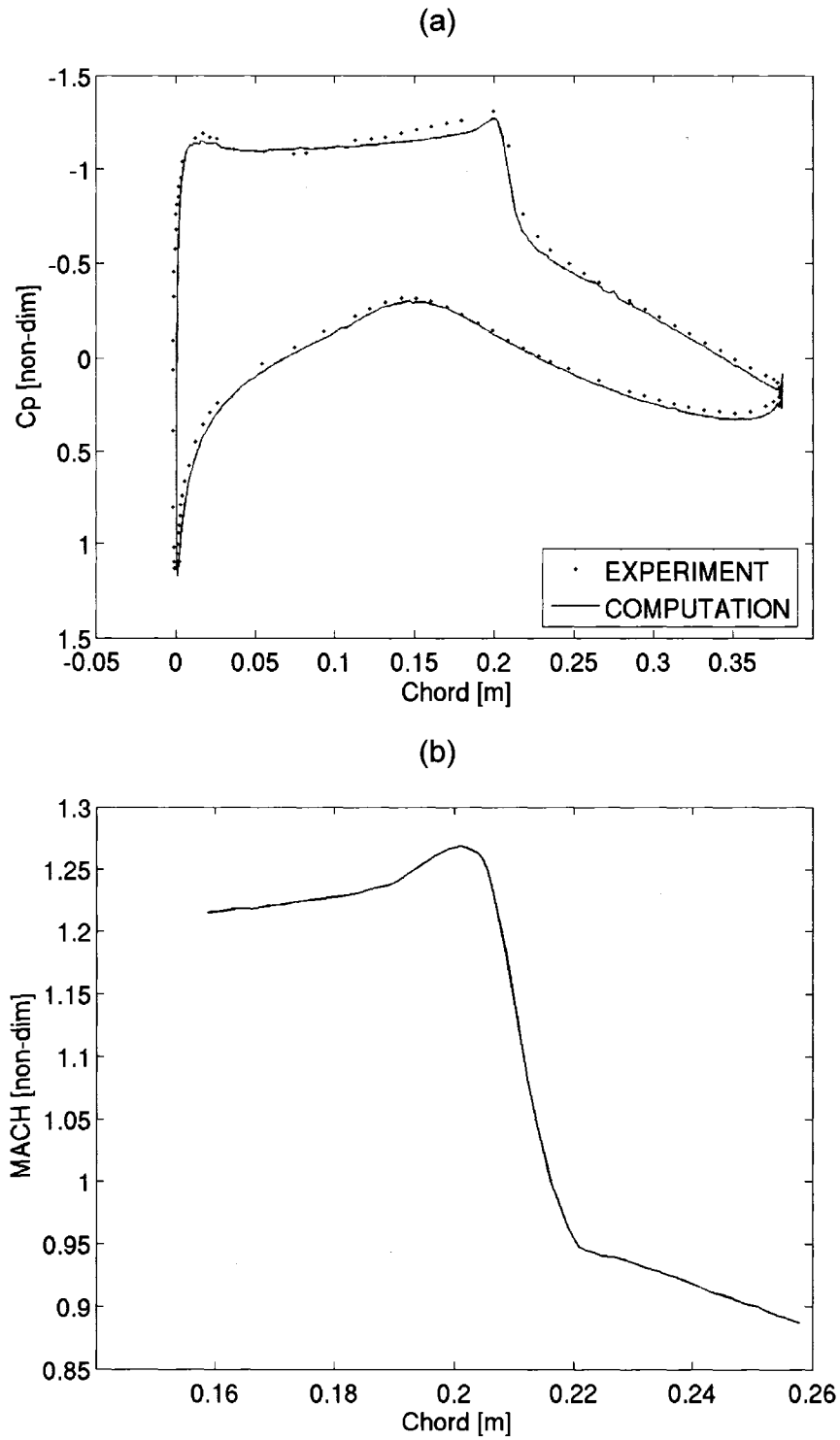


Figure 4-8: RAE 2822 transonic viscous solution using tetra mesh and SU fully second order AV scheme. (a) Wall Coefficient of Pressure distribution – experimental and present solution data. (b) Mach number variation through the shock wave.

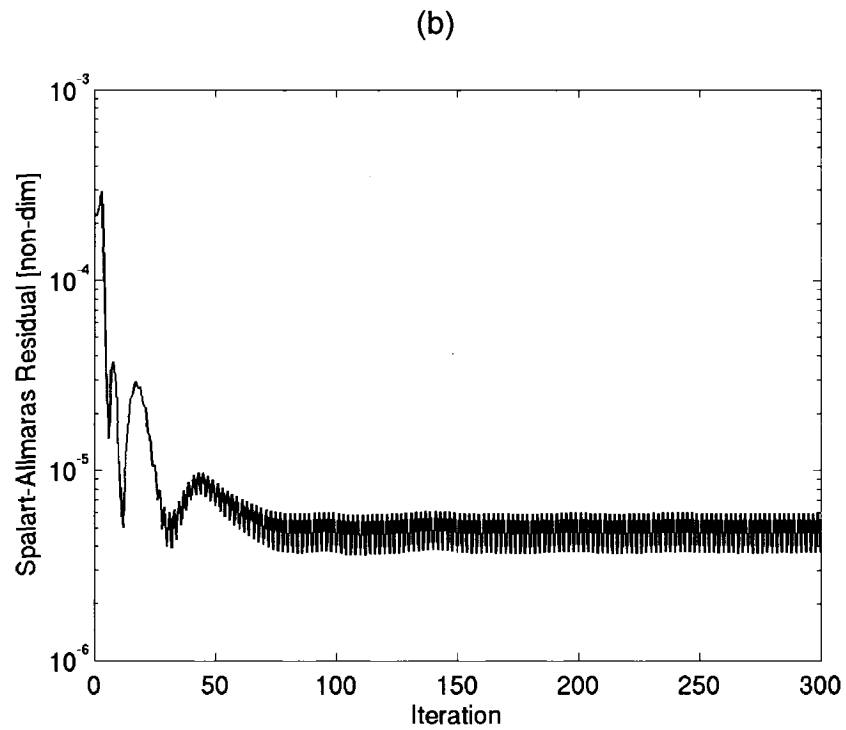
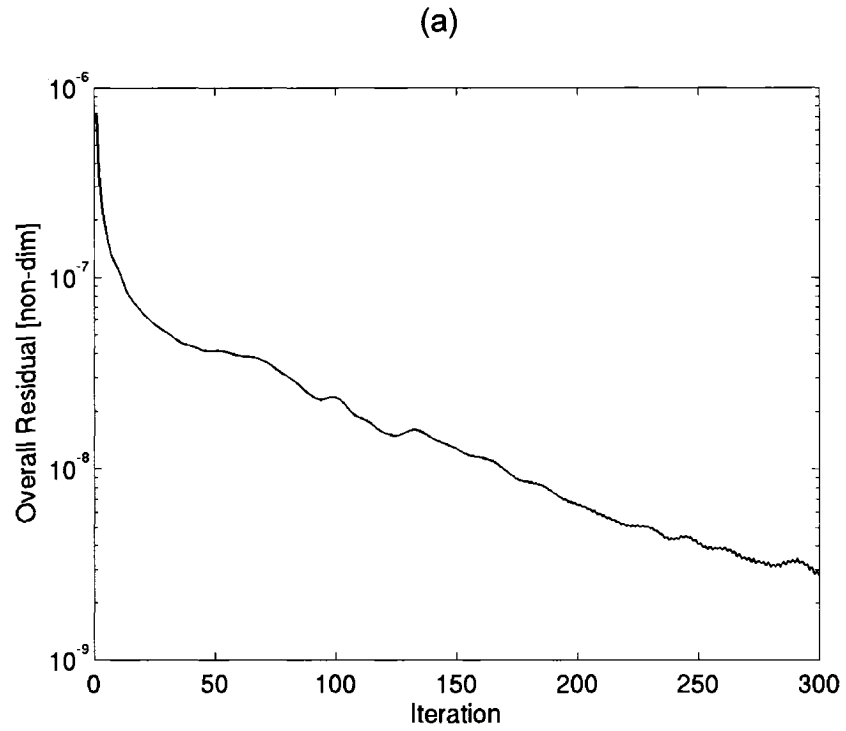


Figure 4-9: Histograms - RAE 2822 transonic viscous solution using tetra mesh and SU fully second order AV scheme. (a) Overall residual. (b) Spalart-Allmaras residual.

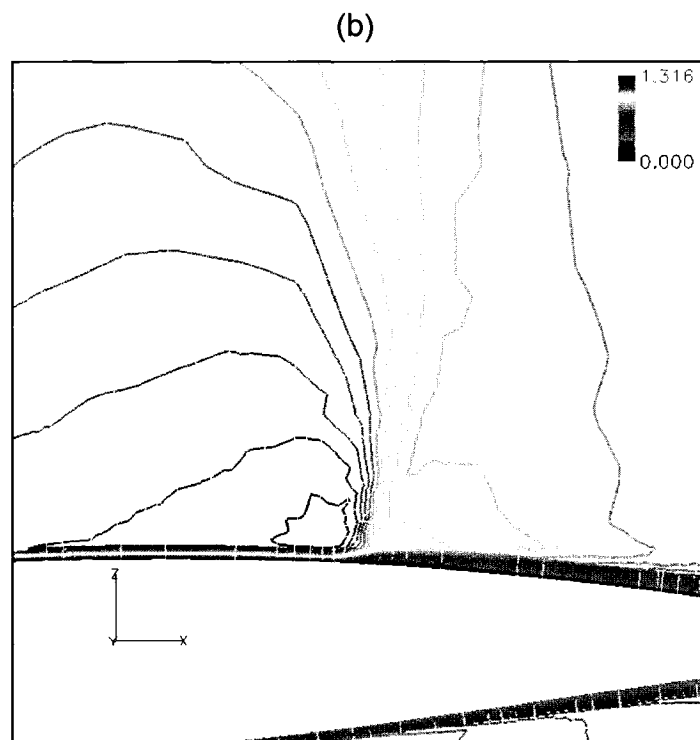
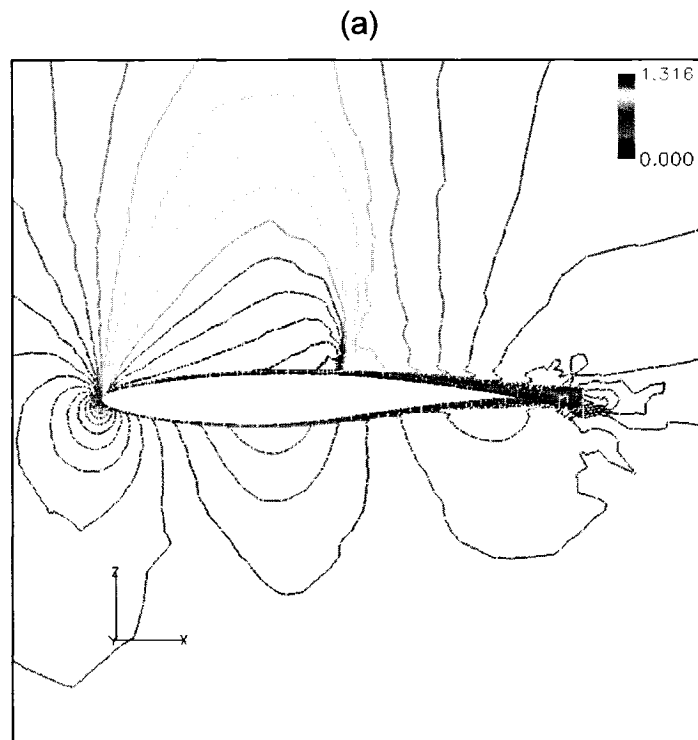


Figure 4-10: (a) Mach contours of the RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme. (b) Zoom in the shock region of (a).

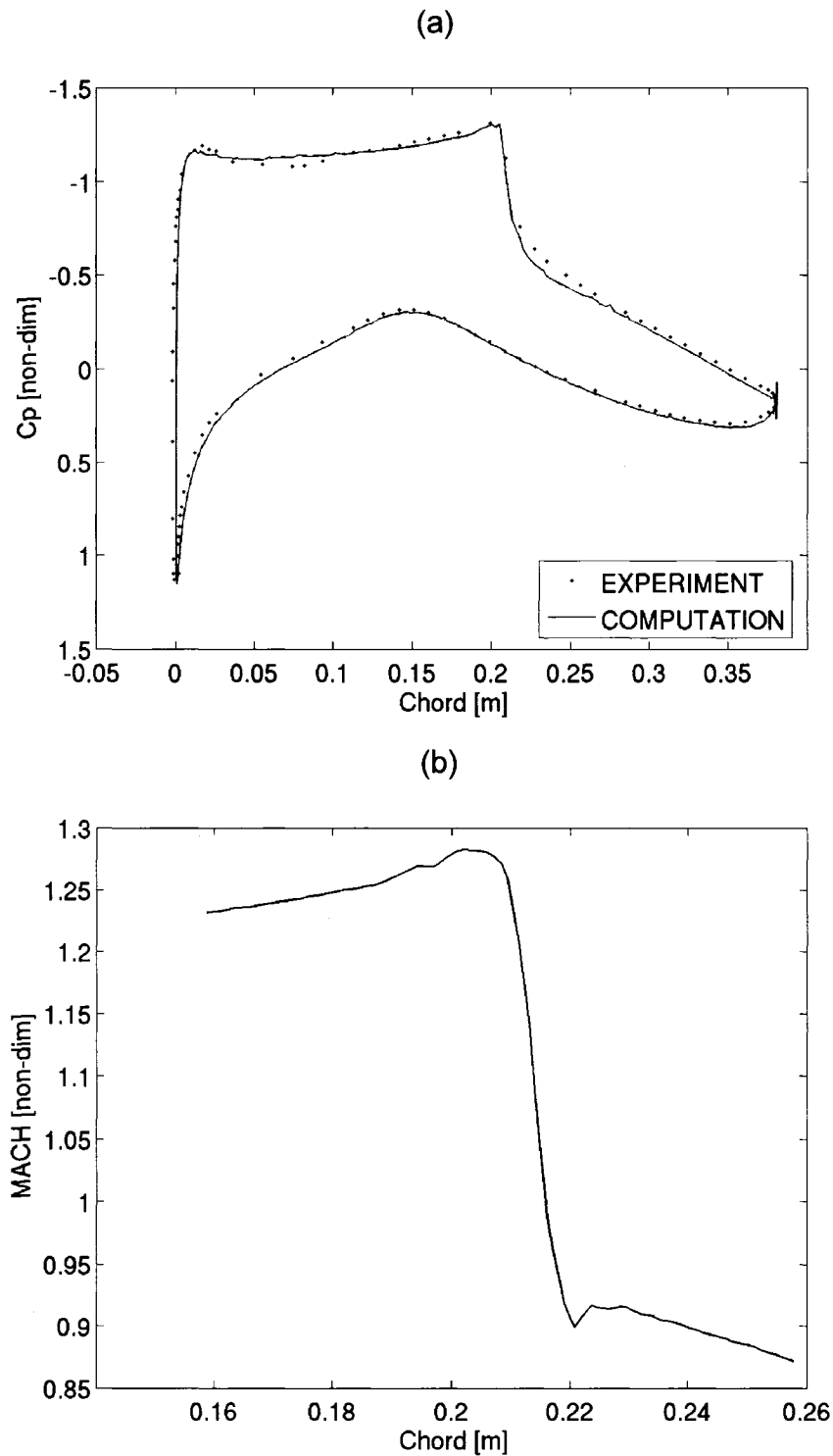


Figure 4-11: RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme. (a) C_p distribution at the wall – present solution against experimental data. (b) Mach number variation through the shock wave.

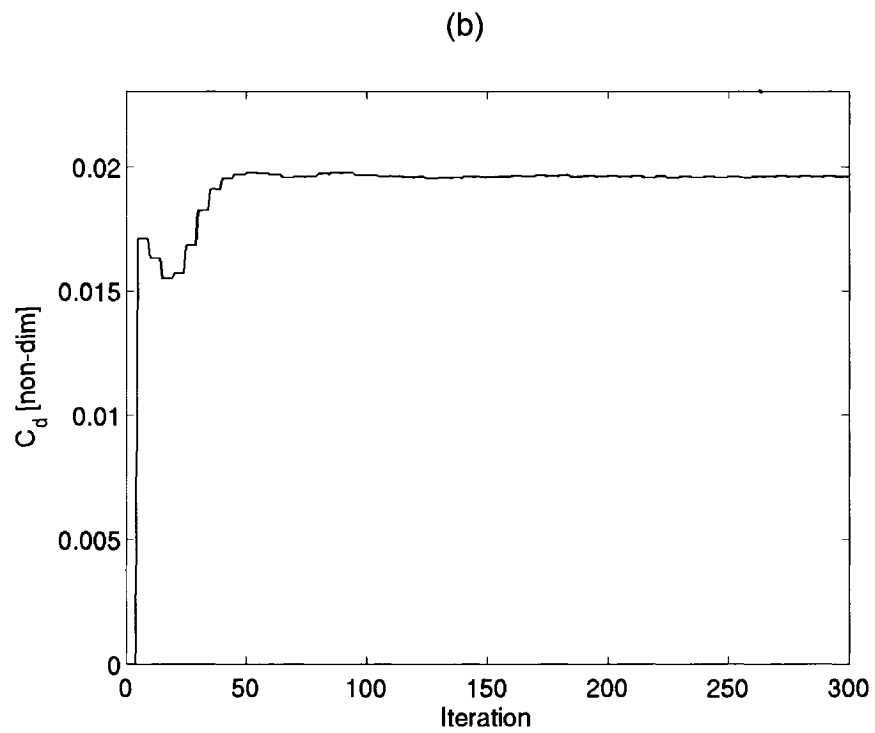
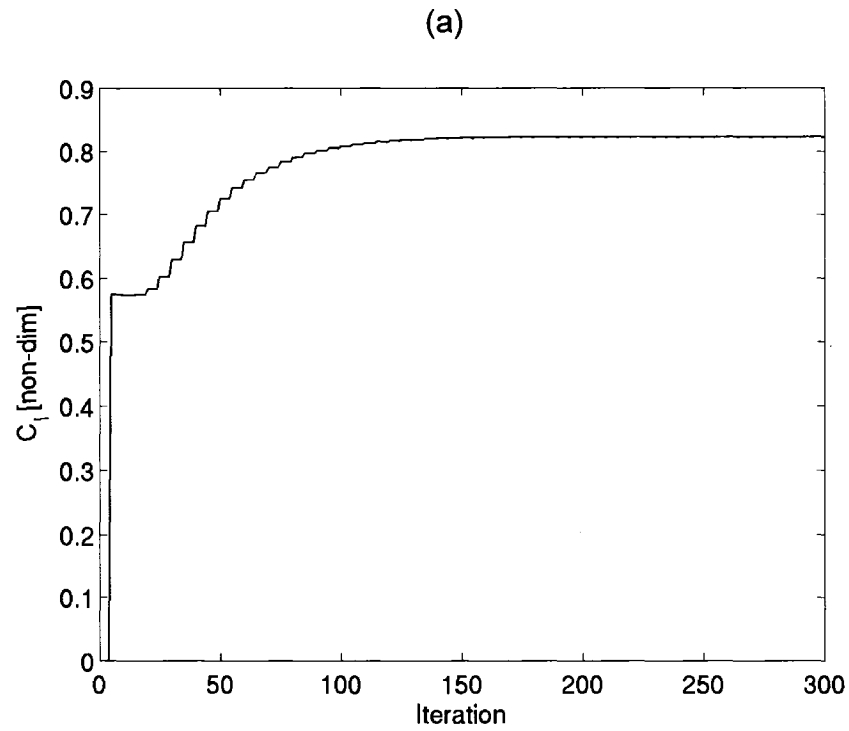


Figure 4-12: RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme - Histograms: (a) Coefficient of lift C_l . (b) Coefficient of drag C_d .

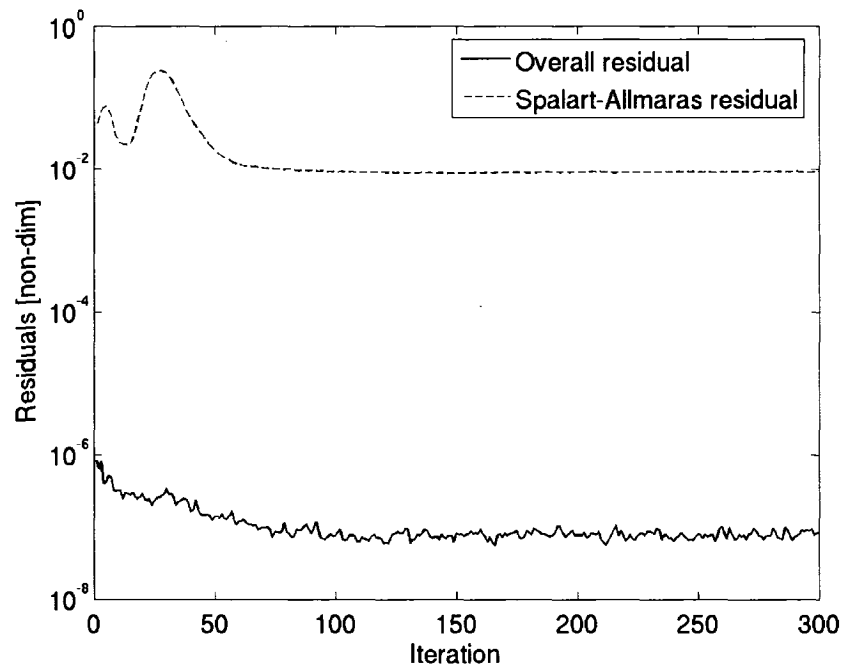


Figure 4-13: RAE 2822 transonic solution obtained using a tetra mesh and SU + shock sensor AV scheme - Histograms for Overall residual and Spalart-Allmaras residual.

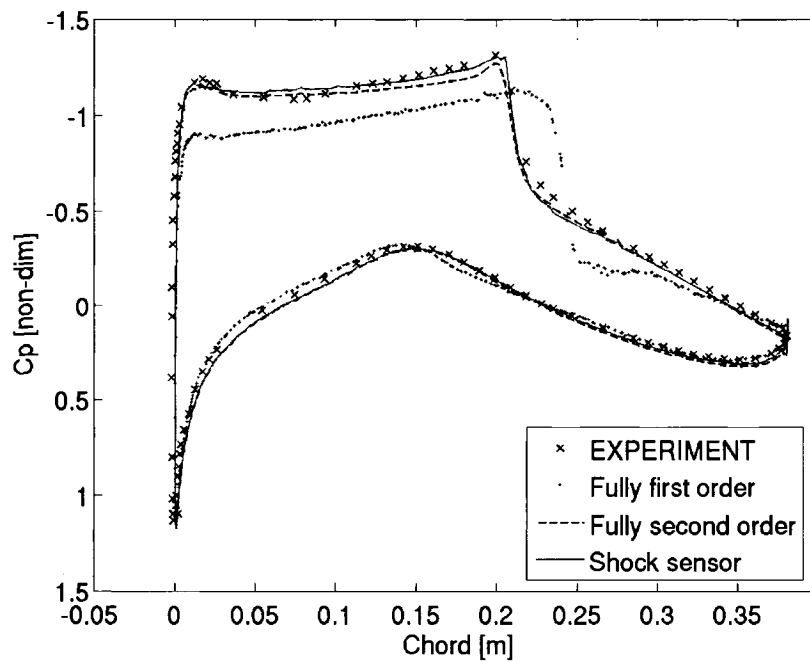


Figure 4-14: C_p comparison for SU fully first order, SU fully second order and SU + shock sensor.

5. Numerical Results - ONERA M6 wing test case

The ONERA M6 wing is a classic CFD validation case for external flows. It has almost become a standard for CFD codes because of its inclusion as a validation case in numerous CFD papers over the years. In the proceedings of a single conference, the 14th AIAA CFD Conference (1999), the ONERA M6 wing was included in 10 of the approximately 130 papers! We thus chose ONERA M6 wing to validate the shock detection implementation code for a 3D viscous flow.

5.1. Wing's Geometry

The dimensions used to define the wing for the present solution are given in the **Table 5-1**. The root airfoil data and the picture of the wing in the wind tunnel can be found in Appendix B. The computation domain stretches approximately 30 root chord lengths above and below the wing's surface, in front of the leading edge and after the trailing edge and in the span direction starting from the symmetry plane.

Wing planform	Swept back
Aspect ratio	3.8
Leading-edge sweep	30.0 degrees
Trailing-edge sweep	15.8 degrees
Taper ratio	0.562
Mean aerodynamic chord	0.6461 m
Semi span	1.1963 m
Reference area	0.7532 m ²

Table 5-1: ONERA M6 wing reference dimensions

5.2. Flow parameters

Reference conditions were chosen in order to match the test 2308 specified in the experimental data report [5] and they are presented in the **Table 5-2**. The values for the flow solver parameters CFL, AV and GMRES are going to be indicated for each simulation of this chapter.

Mach number (M)	0.8395
Reynolds number (Re)	11.72 million
Angle of Attack (AOA)	3.06 degrees
Angle-of-Sideslip	0.0 degrees
Free stream static pressure	93,993 Pa
Free stream static temperature	288 K

Table 5-2: ONERA M6 wing flight conditions

For the 3D test case, the recommended location of the external boundary is located approximately 25 chord lengths from the wall.

5.3. Computation meshes

Figures 5-1 through **5-5** illustrate the grids used to obtain the final solution. **Figure 5-1** is the original grid used in this simulation, generated using the commercial software ICEM CFD. The meshes in **Figure 5-2** and **Figure 5-3** have been obtained after the first two adaptation cycles, while the meshes in **Figure 5-4** and **Figure 5-5** correspond to the third adaptation cycle. **Table 5-3** shows the mesh statistics after each adaptation cycle.

The recommended OPTIGRID settings used for the adaptation were the target number of nodes and 10 internal iterations. For the first adaptation the target was 1.15 million nodes, the second was 1.1 million and 1 million for the last adaptation. The code attempted to equi-distribute the error through node movement, refinement, coarsening and edge swapping, and the final number of nodes are shown in **Table 5-3**.

The total number of nodes has been reduced by 23% after three adaptation cycles. The meshes used in other computations ([2],[3],[4]) have a number of nodes comparable to the number of nodes in our final mesh obtained after the third adaptation cycle (see **Table 5-4**).

	Initial grid	Cycle 1	Cycle 2	Cycle 3
Total Nodes	1,237,031	813,026	1,043,951	956,663
Total nodes on the wall	88,093	72,745	91,562	71,204
Tetras	2,122,490	558,306	800,562	1,440,847
Prisms	1,656,150	1,367,610	1,721,380	1,338,640

Table 5-3: Meshes properties

	Reference [2]	Reference [3]	Reference [4]	Present mesh
Mesh type	C, structured	C-O, structured	Hybrid, unstructured	Hybrid, unstructured
Number of nodes	316,932	665,856	830,476	956,663

Table 5-4: Comparison between present mesh and other CFD meshes

5.4. Discussion of results

The procedure we used for this 3D test case is similar to what we performed for the 2D test case of Chapter 4 with the exception that now we are going to use another tool, OPTIGRID (see also Chapter 3.6). The adaptation strategies of OPTIGRID are the following [9]:

- Moving nodes – equi-distribute the error throughout the domain by moving the position of the grid points.
- Refinement – reduce the error throughout the domain by adding new grid points where the error is higher than a target error threshold.

- Coarsening – equi-distribute the error throughout the domain by removing grid points where the error is lower than the target error threshold.
- Edge swapping – reconnect edges to optimize their orientation and to better align the grid to uni-directional flow features.

The goal is to minimize and make the error uniform everywhere, while maintaining an acceptable number of grid points. Node movement is the only continuous operation and it may be viewed as the driving force of mesh adaptation. Refinement, coarsening and edge swapping are binary (yes/no) operations that complement the action of node movement and should be viewed as a way to accelerate convergence to an optimum grid. In detail, the description of the mesh adaptation procedure is given in **Figure 5-6**. For example, the first adaptation cycle starts from an original ICEM CFD generated mesh, and the corresponding solution generated by FENSAP. Then, OPTIMESH is executed and the outcome is a newly generated mesh (the adapted one) and an interpolated solution for the new mesh obtained from the initial solution. FENSAP is restarted with the newly adapted mesh and the interpolated solution as the initial condition to obtain the final solution.

If the solution after one adaptation cycle is not accurate enough, then the latest solution will be used to make another adaptation cycle as described by the flowchart of **Figure 5-6**. The measure of accuracy of the solution is often either the lift or drag coefficient, depending on the flight condition. In the case of a cruise flight condition, drag coefficient is the preferred measure of accuracy. At take-off and landing lift coefficient is monitored. The adaptation cycle is halted when global convergence of the lift or drag coefficient is achieved. This often requires at least a three to four decimal place accuracy of either the lift or drag coefficient. The ONERA M6 test case required three adaptation cycles to attain global convergence.

The values for parameters G and S determined for the 2-D case RAE 2822 are used for the 3-D test case ONERA M6 wing. Just parameter F is modified from $F=8$ in 2D test case to $F=4$ in 3D test case.

Through numerical experiments (see Chapter 4) it has been determined that values as $G=1.4$ and $S=0.95$ can be set as default values, while the parameter F can be varied in order to achieve the desired accuracy of the solution. The parameter F will have a dominant effect on the shock capturing and its value will be indicated in each of the following simulations.

5.4.1. Initial solution

The initial solution was obtained through several parametric cycles, through the restart procedure described in Chapter 4. At each parametric cycle, a new AV coefficient is selected and approximately 200 iterations are performed. Therefore, using the parameter $F=1$ the solution is obtained after 1,332 iterations, and an average time of 30 seconds per iteration was needed, using 32 CPUs. This solution has been generated to serve as the initial condition for the next simulation, where the parameter F will be used to control the amount of diffusivity in the system based on the computed shock sensor. This solution is robust, but as we expected the shock wave on the upper surface is not well captured. Due to the high amount of diffusivity in the solution the shock is smeared, as shown in **Figure 5-7** and **Figure 5-8**.

The convergence of the solution is illustrated by the Cl and Cd histograms, while the overall residual drops three orders of magnitude, reaching an order of magnitude of 10^{-07} . Due to the diffusive character of the solution, the pressure distribution does not match the experimental data, and as a consequence the lift and drag coefficients are inaccurate and the drag is too large due to the artificial viscous contribution. In the experiment, the wing section corresponding to 80% of the wing semi span features two shock waves, one closer to the LE and the other positioned towards the middle of the wing,[5].

Neither shock is properly captured in the initial solution; our goal is to modify the parameter in the AV allocation logic for proper shock capturing. Parameter F will be increased, allowing less AV for the momentum equation when we are in the first order dominated region of the domain. The amounts of AV applied in each parameter cycle of this initial solution are indicated in **Table 5-5**. The start value is 10^{-04} , while the end value is the lowest possible in FENSAP. An AV coefficient of 10^{-10} will be maintained in the following simulations, but the amount of AV that corresponds to the momentum equation will be progressively decreased by the means of parameter F.

Parametric cycle #	AV coefficient
1	0.0001
2	1e-05
3	1e-06
4	1e-07
5	1e-08
6	1e-09
7	1e-10

Table 5-5: AV coefficients corresponding to each parametric cycle - initial solution.

5.4.2. Restart solution before mesh adaptation

We now restart based on the previously obtained solution, and try to decrease the AV in the momentum equation by modifying the parameter F. Through numerical experiments, a value of $F=4$ was determined to provide a good solution. Below, two types of shock sensors have been tried: one based on the second derivative of pressure, and one based on the second derivative of entropy.

5.4.2.1. Solution obtained using a shock sensor based on pressure

The solution herein was obtained using 32 CPUs, providing a computation speed of approximately 30 seconds per iteration. The other settings were CFL=500, AV=1e-10 and default settings for GMRES. We notice a very small variation in the lift and drag coefficients before they reached steady state, indicating that the initial solution used to restart the present one was not far from the correct one. The shock is captured better this time, given the reduced AV we have been using for the momentum equation and the graphs in **Figure 5-9** and **Figure 5-10** are evidence of this fact.

	Other CFD, [3]	Computation – present solution	Relative error ϵ [%] - eq.(4-1)
Lift Coefficient C_l	0.26806	0.2709	1.06
Drag Coefficient C_d	0.01717	0.01801	4.89

Table 5-6: ONERA M6 – aerodynamic coefficients for intermediate solution.

The aerodynamic coefficients are comparable to values obtained by Kalitzin [3]. However, there still exist oscillations in the solution.

5.4.2.2. Solution obtained using a shock sensor based on entropy

As an attempt to clean up the oscillations from the solution, the shock sensor is calculated now based on the second derivative of entropy. The solution is given in **Figure 5-11** and **Figure 5-12**. The settings are the same as in the pressure based shock detector. However, the differences between pressure and entropy detection solutions are not significant and the oscillations did not disappear. Therefore, the following simulations will be based on the pressure sensor.

5.4.3. Solution after mesh adaptation

The solution shown in previous section and obtained for $F=4$ displays a preliminary agreement with test data that has to be improved, but the presence of the oscillations is undesired and they have to be removed. The method we used to improve the solution is the mesh adaptation and the procedure is

explained in **Figure 5-6**. Through mesh adaptation we expect to improve the solution convergence and obtain a better comparison to the experimental data and other CFD solutions.

The first adaptation cycle solution is shown in **Figure 5-13**. The FENSAP settings are CFL=500, AV=1e-10, default values for GMRES, while the minimum clearance to the wall was set in OPTIMESH at 2.5e-5. We can see an immediate improvement of the solution, both in reduction of oscillations and shock capturing. The shock waves after adaptation are less smeared than before adaptation and at the 80% span station both shocks are sharper.

We proceed now with the second adaptation cycle, as shown in **Figure 5-14**. The settings for the flow solver and mesh adaptation are kept the same as in the first adaptation cycle, except the target number of nodes. As we are not after a mesh size reduction, the target number of nodes influence is not going to be discussed herein. The shock waves are now better captured and the experimental data is better matched. The single problem remained to be solved being the presence of oscillations. This motivates us to attempt a third adaptation cycle.

The settings of this third adaptation cycle are CFL=150, AV=1e-10, GMRES =160 for turbulence equation and =375 for the momentum equation, leading to an average time of 4 minutes and 30 seconds per iteration, using 16 CPUs. The simulation required 120 iterations and a convergence of three order of magnitude drop with an order of magnitude of 10^{-08} has been achieved (see **Figure 5-15** to **Figure 5-23**).

The Cp contours of this solution show that the oscillations have been removed by the third adaptation cycle. Based on the coefficient of drag convergence criterion we can retain this solution as the final one for this transonic viscous flow simulation (see **Table 5-7**).

	Initial solution	Solution after first adaptation cycle	Solution after second adaptation cycle	Solution after third adaptation cycle
Lift Coefficient C_l	0.26722	0.27151	0.26967	0.26213
Drag Coefficient C_d	0.01994	0.01865	0.01733	0.01664

Table 5-7: Aerodynamic coefficients values for each solution

The initial solution for this test case was obtained for a grid of approximately 1.2 million mesh points, but was affected by oscillations. After two adaptation cycles the solution improved, but still showed oscillations. We stopped the adaptation after the third adaptation cycle because the oscillations were almost completely removed and the shock capturing was good. Also, the C_d decreased from one adaptation cycle to the other showing signs of convergence – **Table 5-8**.

	Between initial solution and first adaptation cycle	Between first and second adaptation cycles	Between second and third adaptation cycles
Relative error - C_l	1.6 %	0.67 %	2.8 %
Relative error - C_d	6.4 %	7.07 %	3.98 %

Table 5-8: Relative error for aerodynamic coefficients

The pressure coefficient distribution at each span location compares very well with other numerical solutions ([2],[3],[4]) and the experimental data. The use of grid adaptation has allowed the double shock at the 80% span station to be captured.

Determining the cause of differences between the computation and experiment requires sensitivity studies with respect to such things as the grid, turbulence model, and algorithms. Refining the resolution of the boundary layer may improve comparisons. Further stream wise refinement of the grid would help capture the shocks on the upper surface and reduce small oscillations at the TE. The highest Mach number on the wing is obtained at the tip of the wing, close to

the LE. This value is slightly larger than normal and probably is generated in conjunction with the mesh size in that region.

The lambda shock wave obtained in the present numerical simulation of the transonic viscous flow around ONERA M6 wing presents the interaction of two shock waves that appear on the wing's upper surface. One of them is situated on the wing's upper surface in the vicinity of the LE and the other traverses diagonally the middle of the upper surface of the wing. The waves seem to have a similar intensity before the joining region, situated at about 80% of the wing semi span. After they coalesce, a stronger single shock wave is formed that continues towards the tip of the wing. Such a strong shock constitutes an important contribution in the overall drag coefficient presented in the **Table 5-9**.

	Other CFD, [3]	Computation – present solution	Relative error ε [%] - eq.(4-1)
Lift Coefficient C_l	0.26806	0.2621	2.22
Drag Coefficient C_d	0.01717	0.01664	3.09

Table 5-9: Aerodynamic coefficients for 3rd adaptation cycle solution.

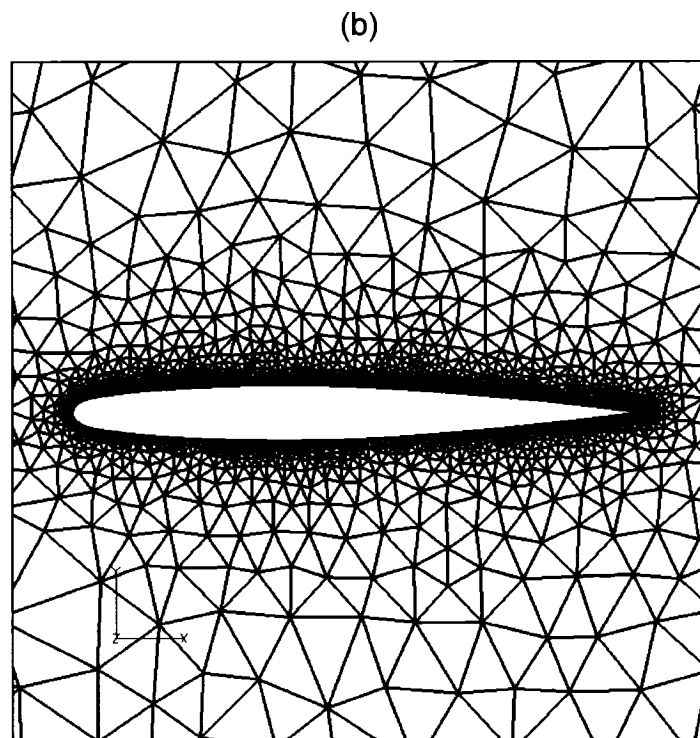
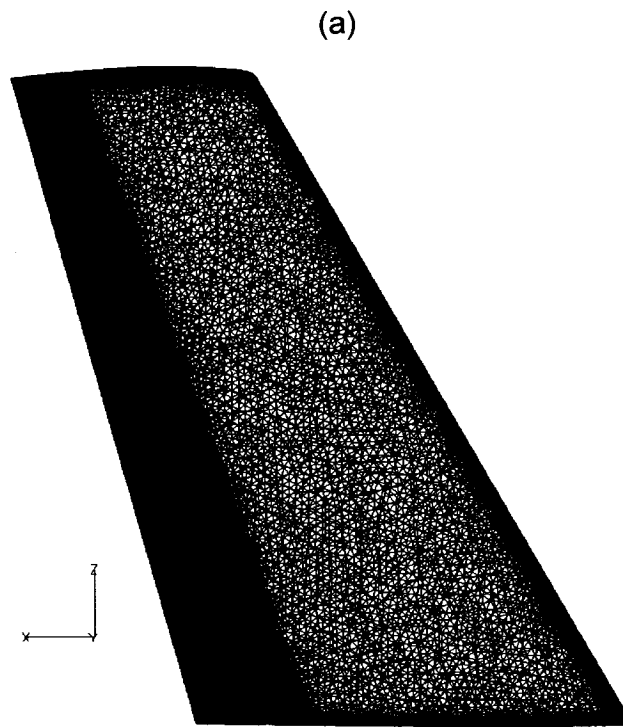


Figure 5-1: Initial mesh (i.e. before adaptation) (a) Upper surface view. (b) Wing root detail.

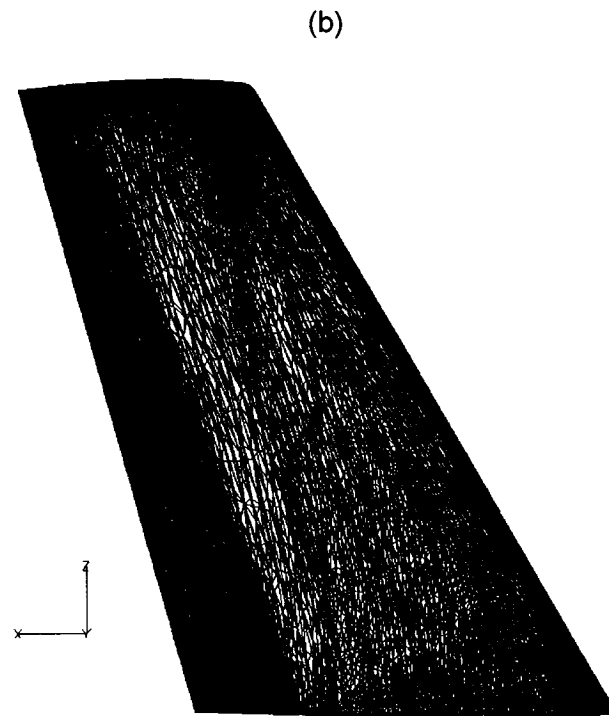
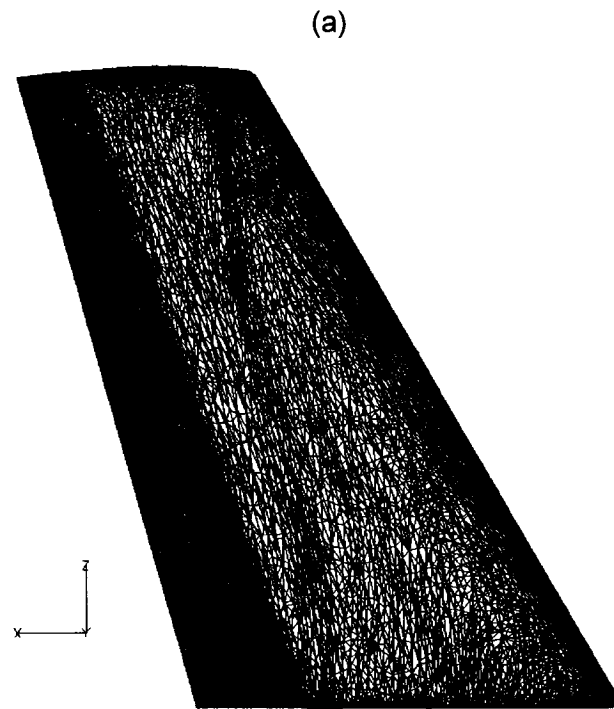
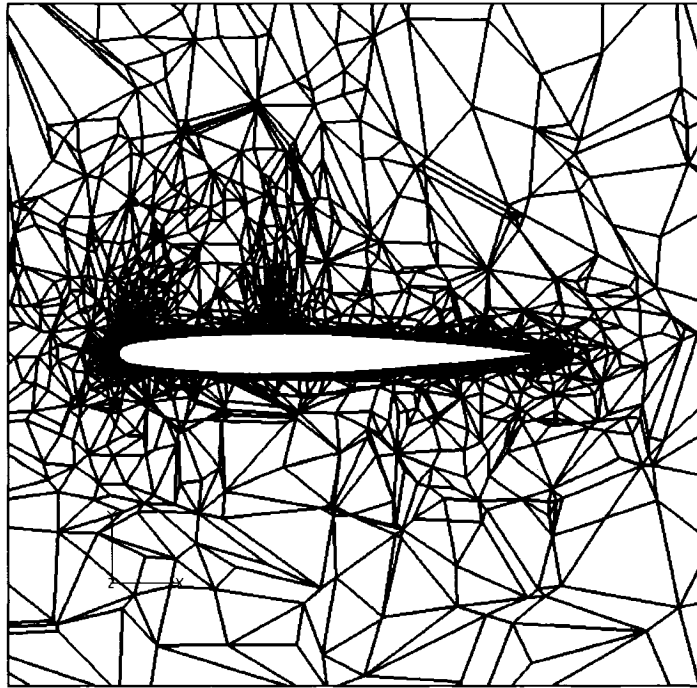


Figure 5-2: (a) Top view of the mesh obtained after 1st adaptation. (b) Top view of the mesh obtained after 2nd adaptation.

(a)



(b)

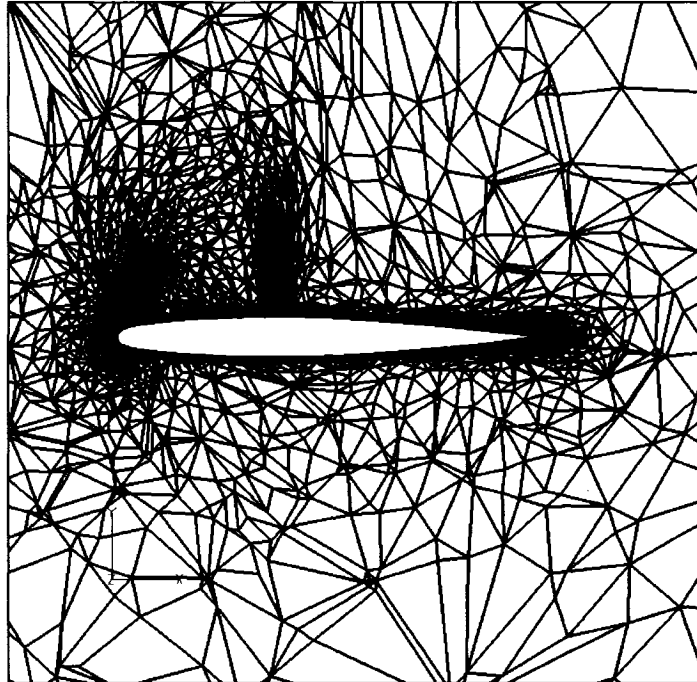


Figure 5-3: (a) Mesh obtained after 1st adaptation – view at 80% wing semi span. (b) Mesh obtained after 2nd adaptation – view at 80% wing semi span.

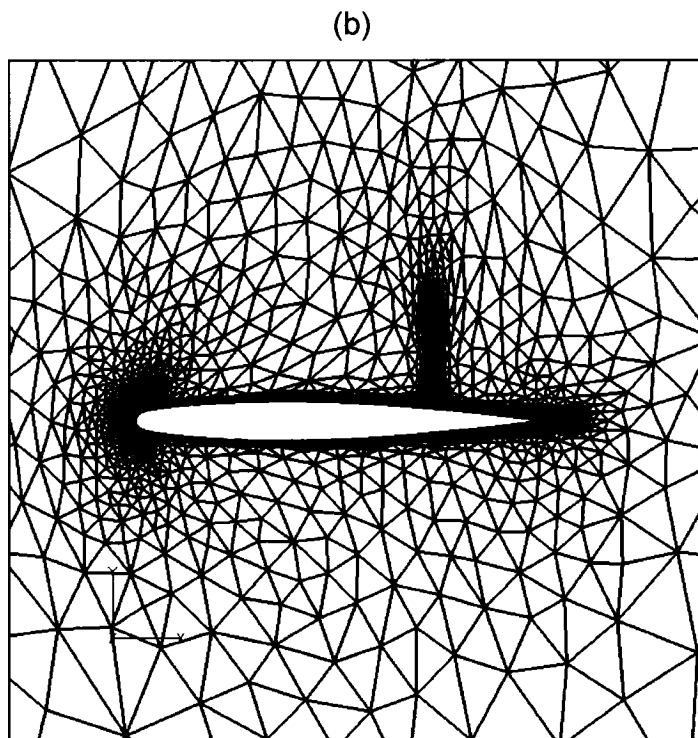
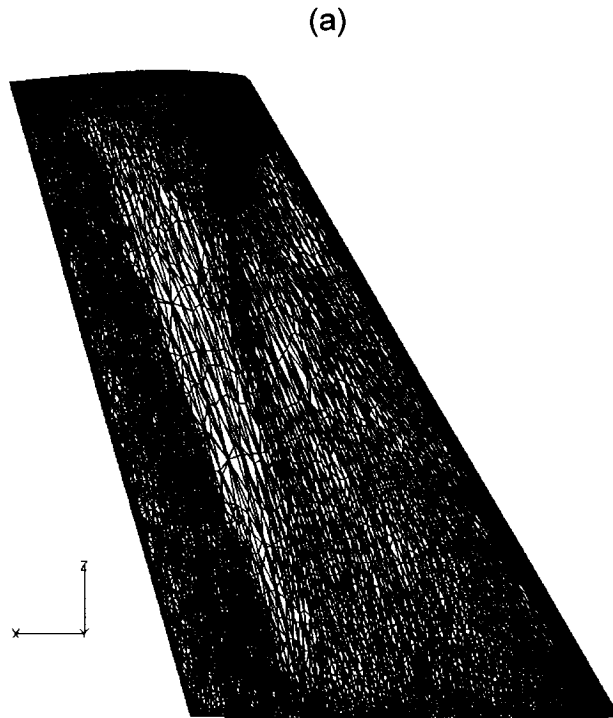
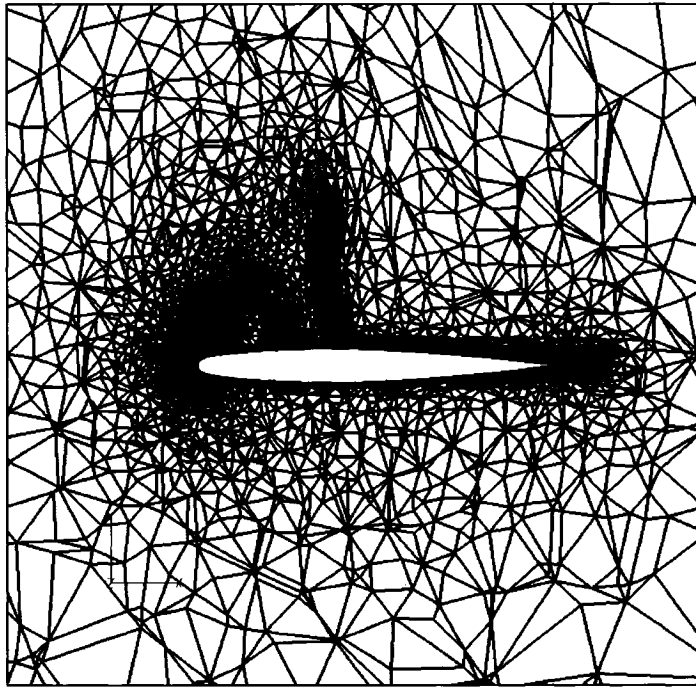


Figure 5-4: (a) Top view of the mesh obtained after 3rd adaptation. (b) Mesh obtained after 3rd adaptation – view at the wing root.

(a)



(b)

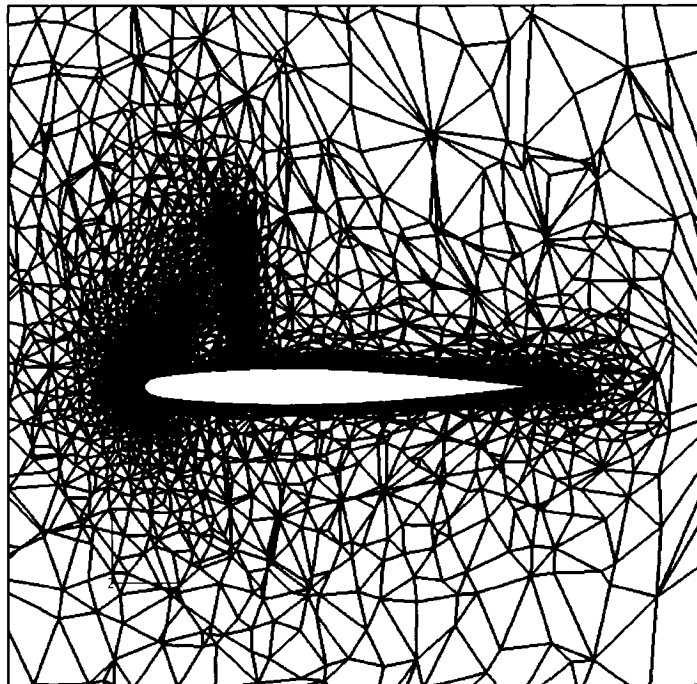


Figure 5-5: (a) Mesh obtained after 3rd adaptation – view at 80% wing semi span. (b) Mesh obtained after 3rd adaptation – view at 95% wing semi span.

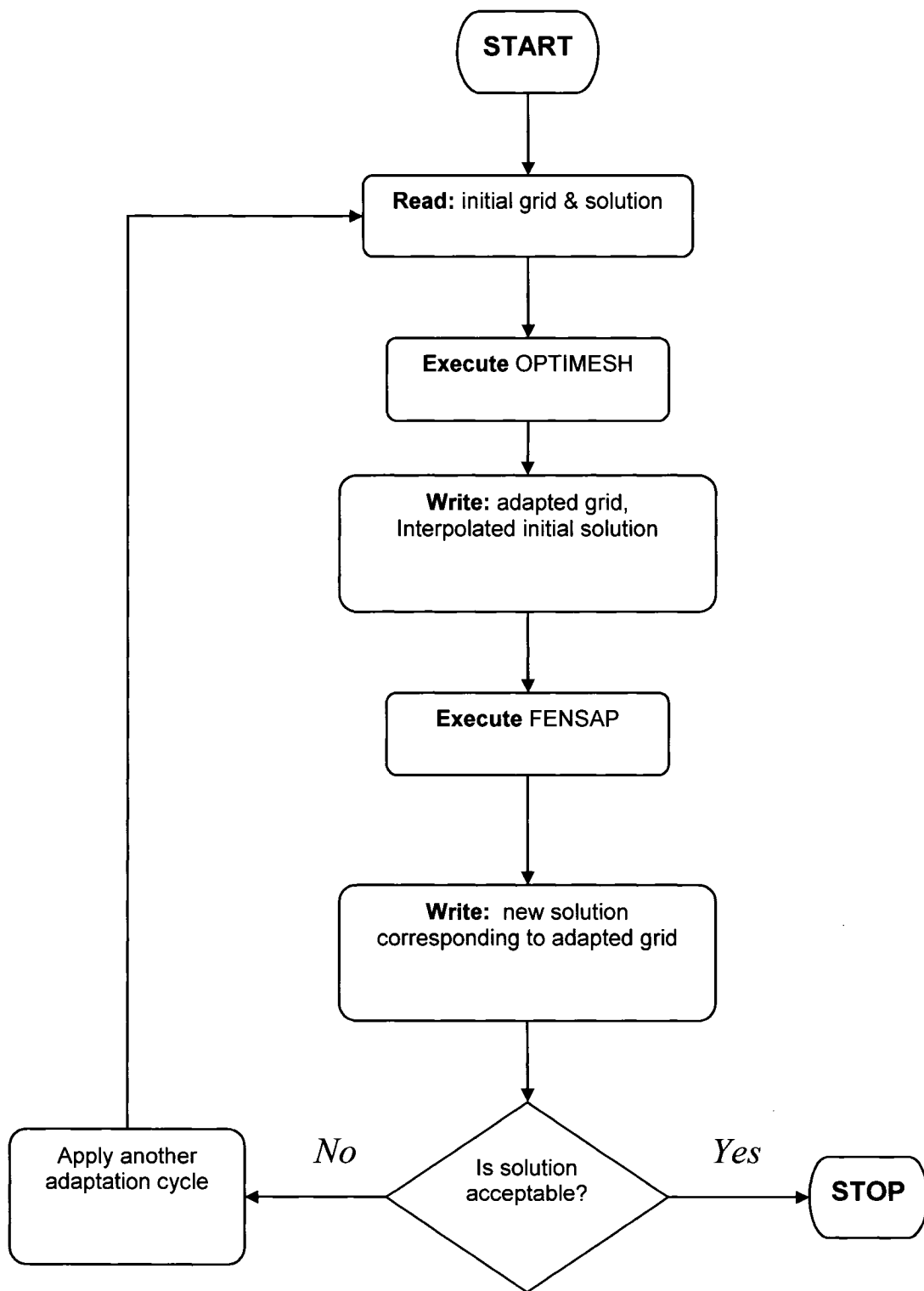


Figure 5-6: Mesh adaptation procedure

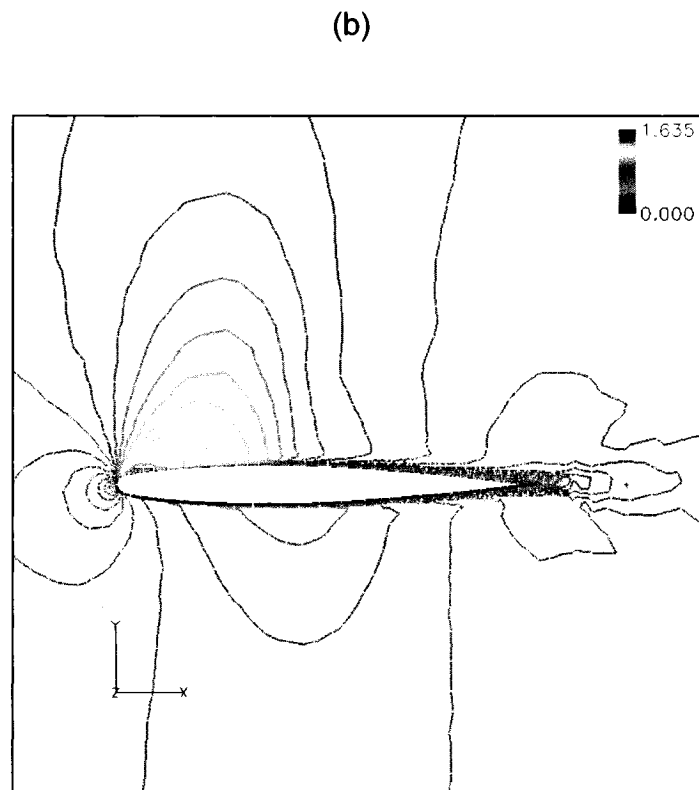
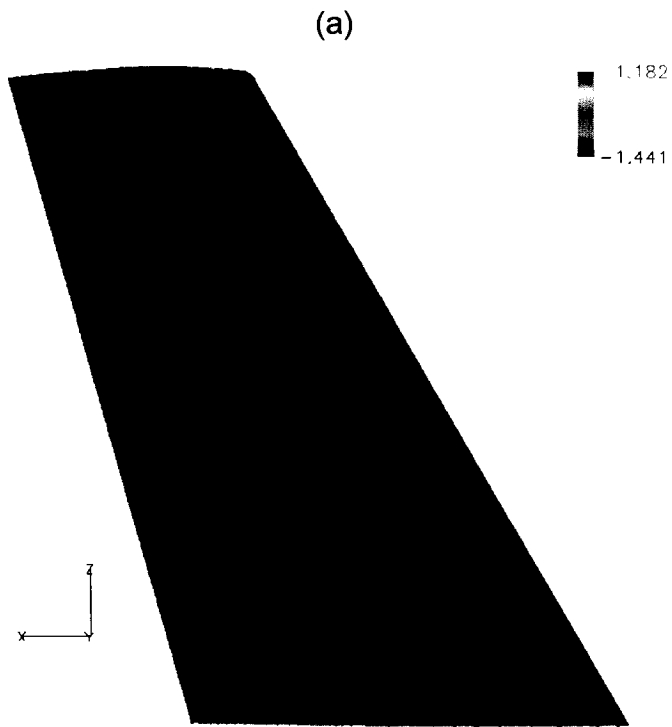


Figure 5-7: Initial solution obtained for $F=1$ (a) C_p distribution on the wing.(b) Mach number contours at 80% of wing semi span.

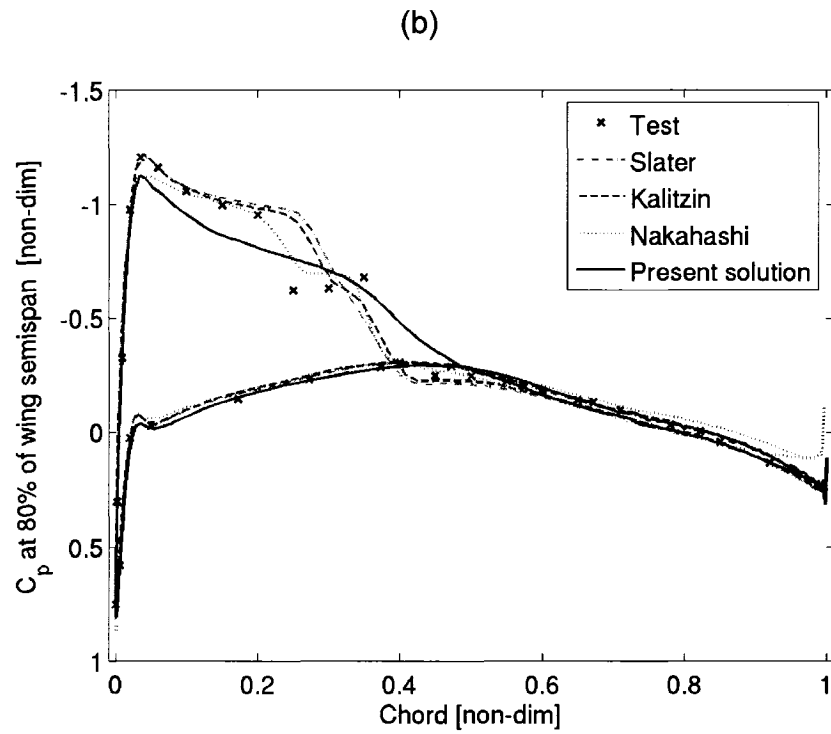
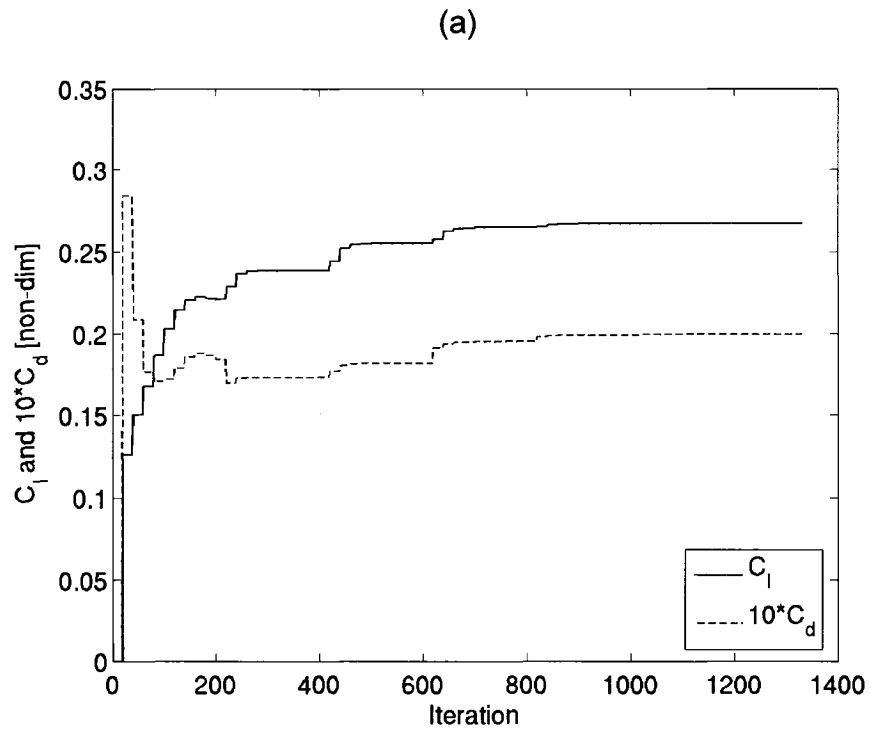


Figure 5-8: Initial solution obtained for $F=1$ (a) C_l and C_d coefficients. (b) Wall pressure coefficient distribution at 80% wing semi span.

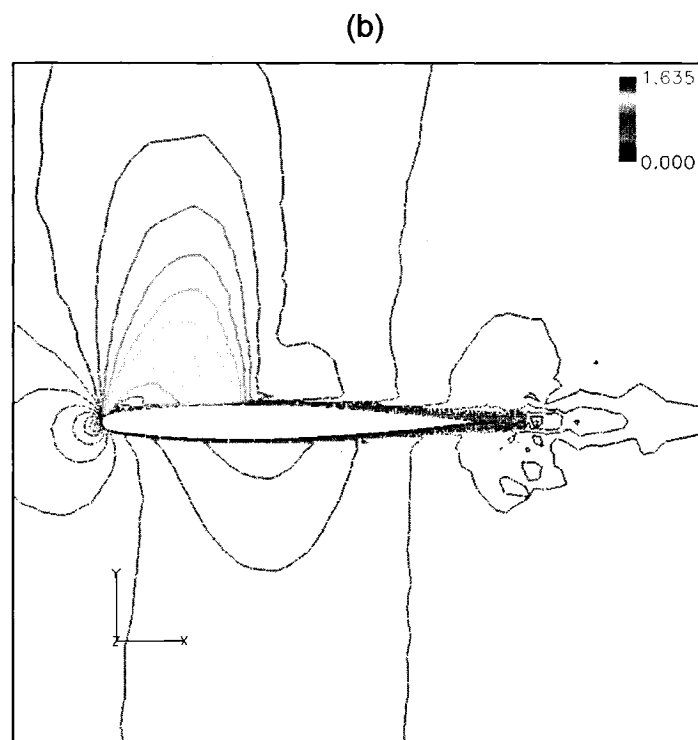
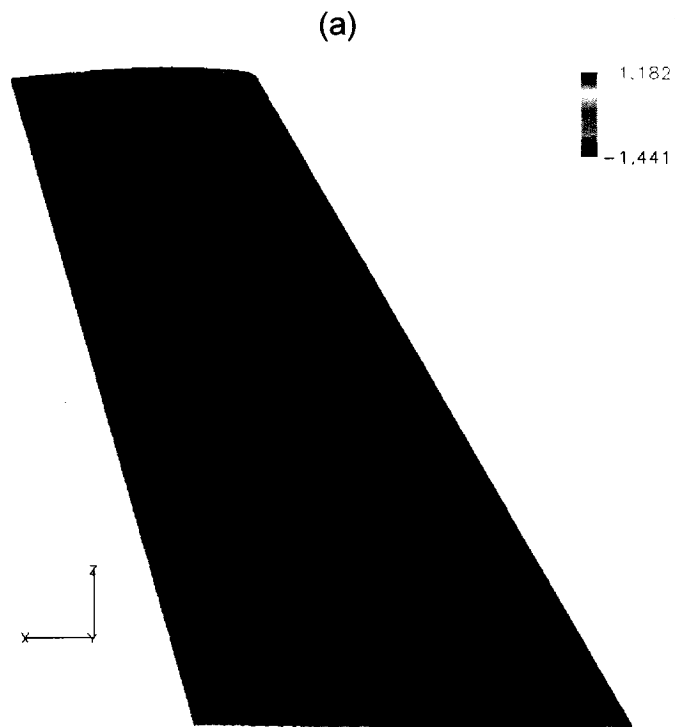


Figure 5-9: Solution obtained for $F=4$, isotropic grid and pressure shock sensor.
 (a) C_p distribution on the wing – top view. (b) Mach contours at 80% wing semi span.

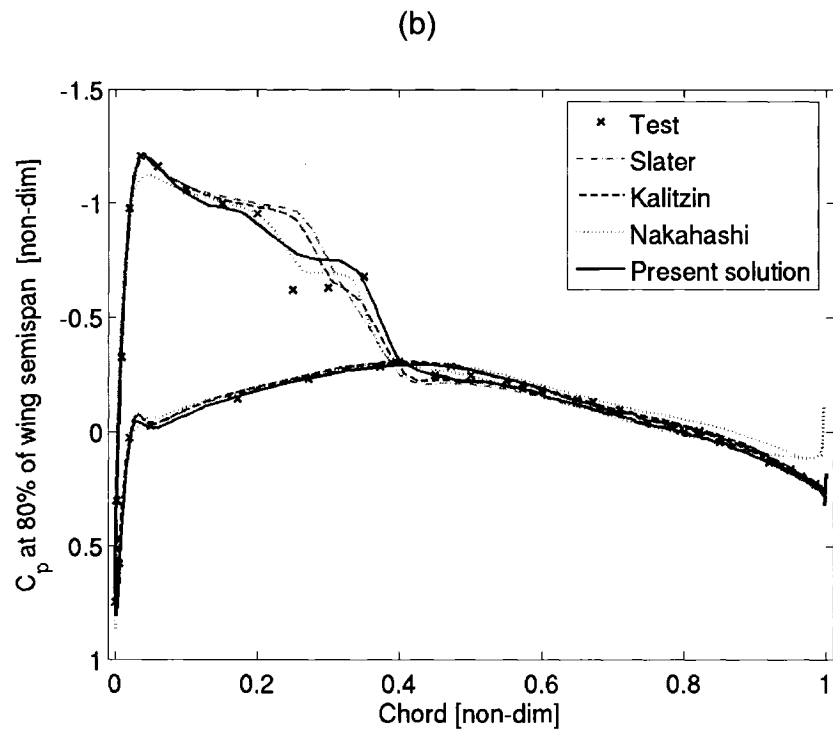
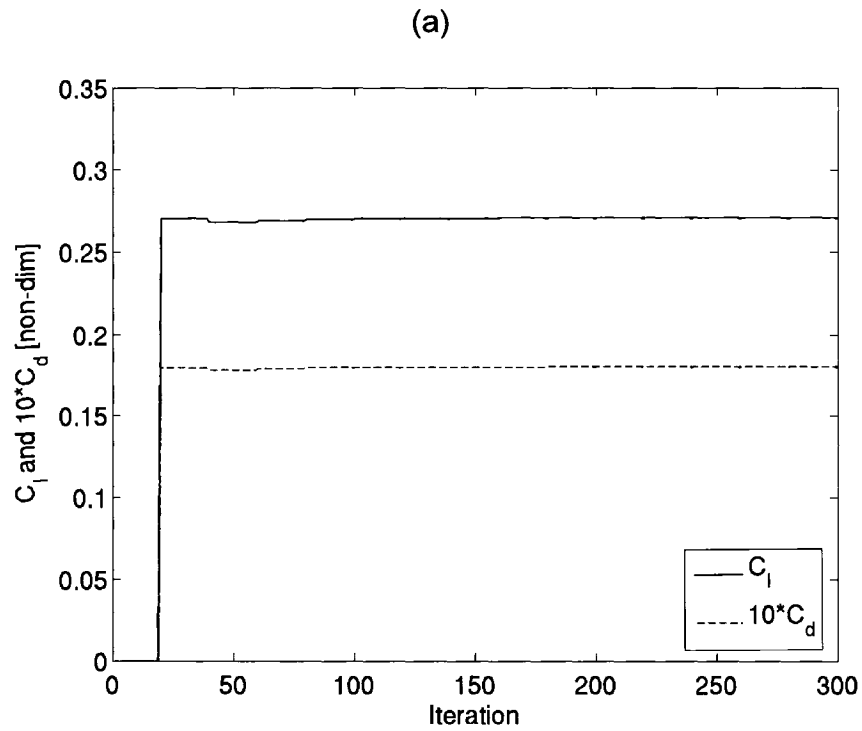


Figure 5-10: Solution obtained for $F=4$, isotropic grid and pressure shock sensor. (a) C_l and C_d histograms. (b) C_p distribution comparison.

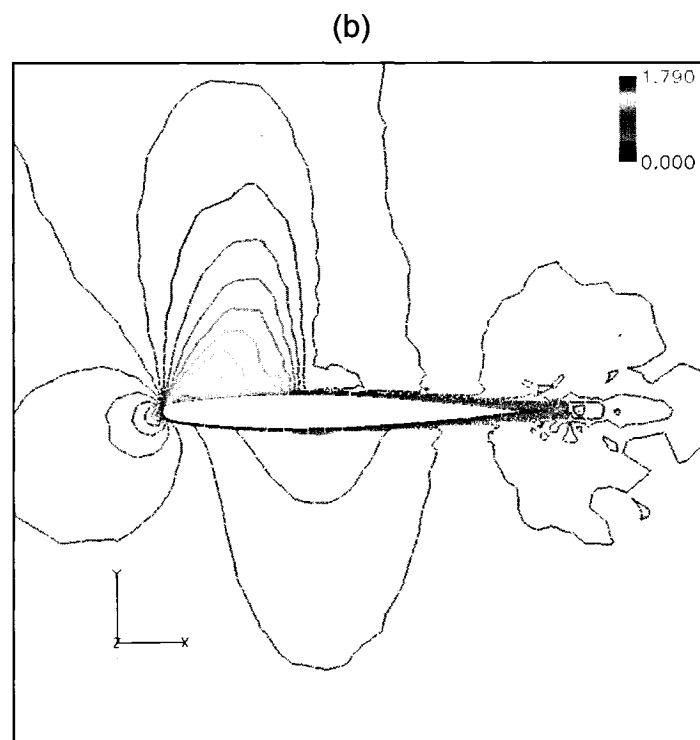
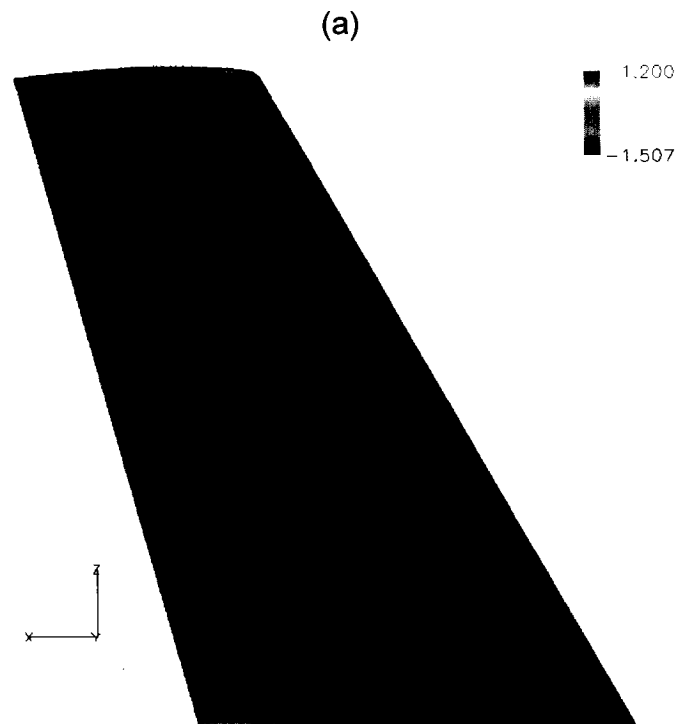


Figure 5-11: Solution obtained for $F=4$, isotropic grid and entropy shock sensor.
 (a) C_p contours on the wing – top view. (b) Mach contours at 80% of the wing semi span.

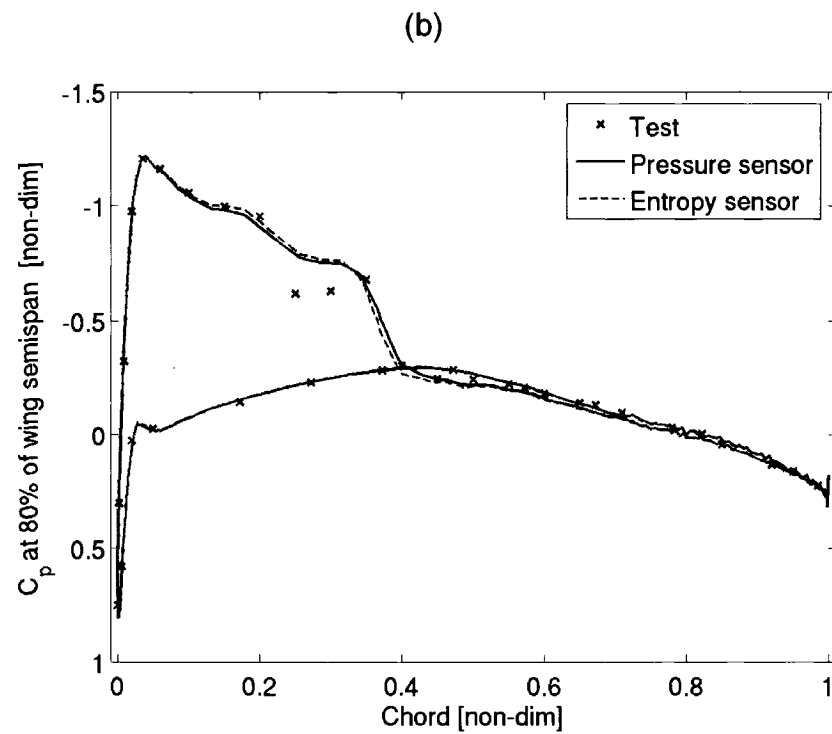
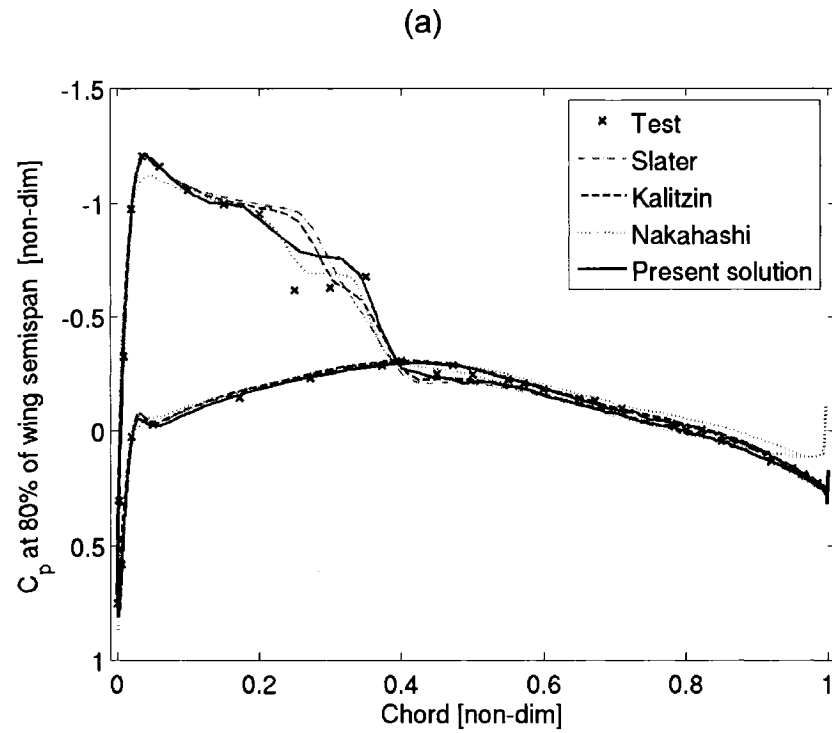


Figure 5-12: (a) Solution obtained for $F=4$, isotropic grid and entropy shock sensor - C_p distribution comparison. (b) C_p comparison between the two shock sensor types.

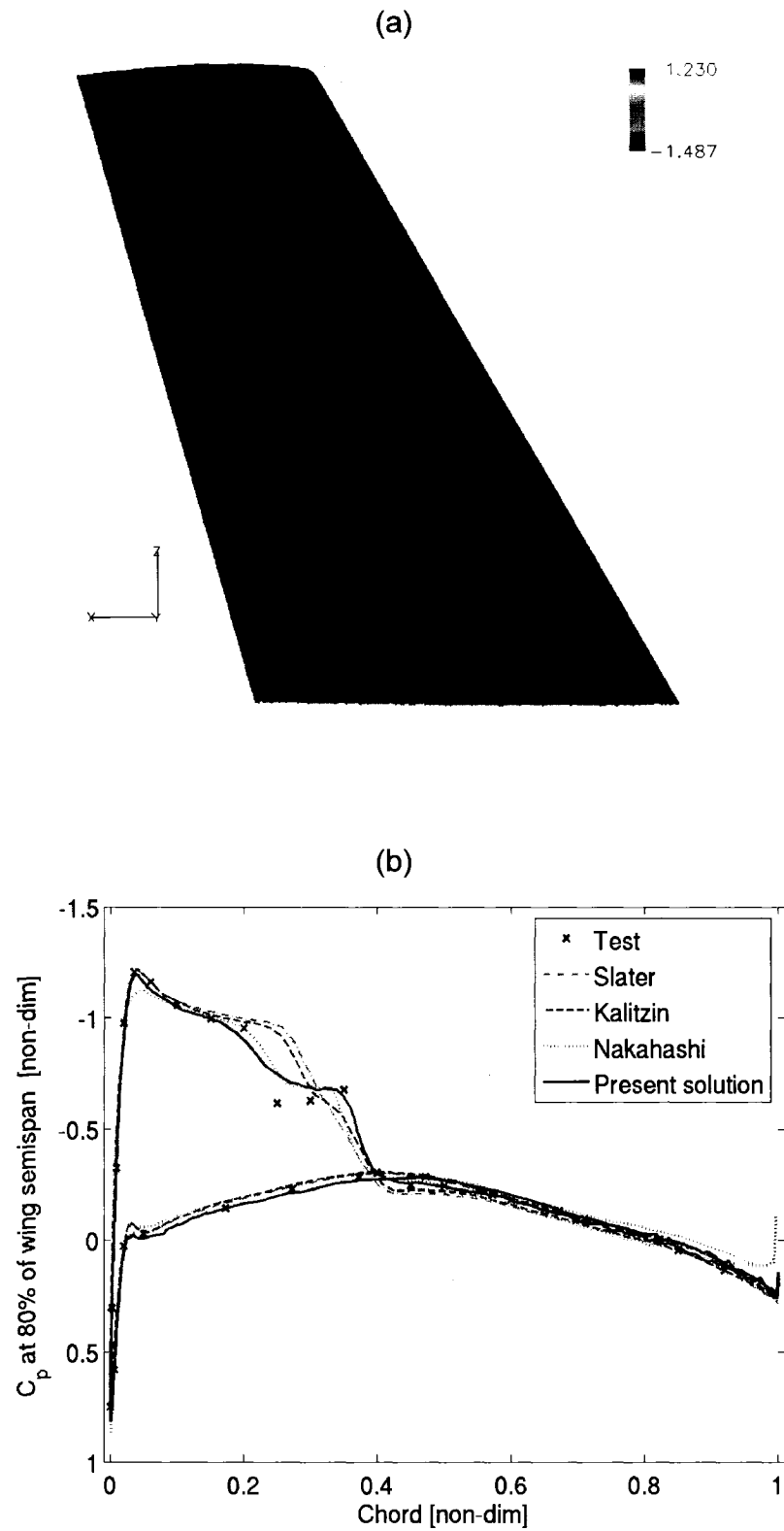


Figure 5-13: 1st adaptation cycle solution: (a) C_p contours on the wing – top view. (b) Wall C_p comparison at 80% of wing semi span.

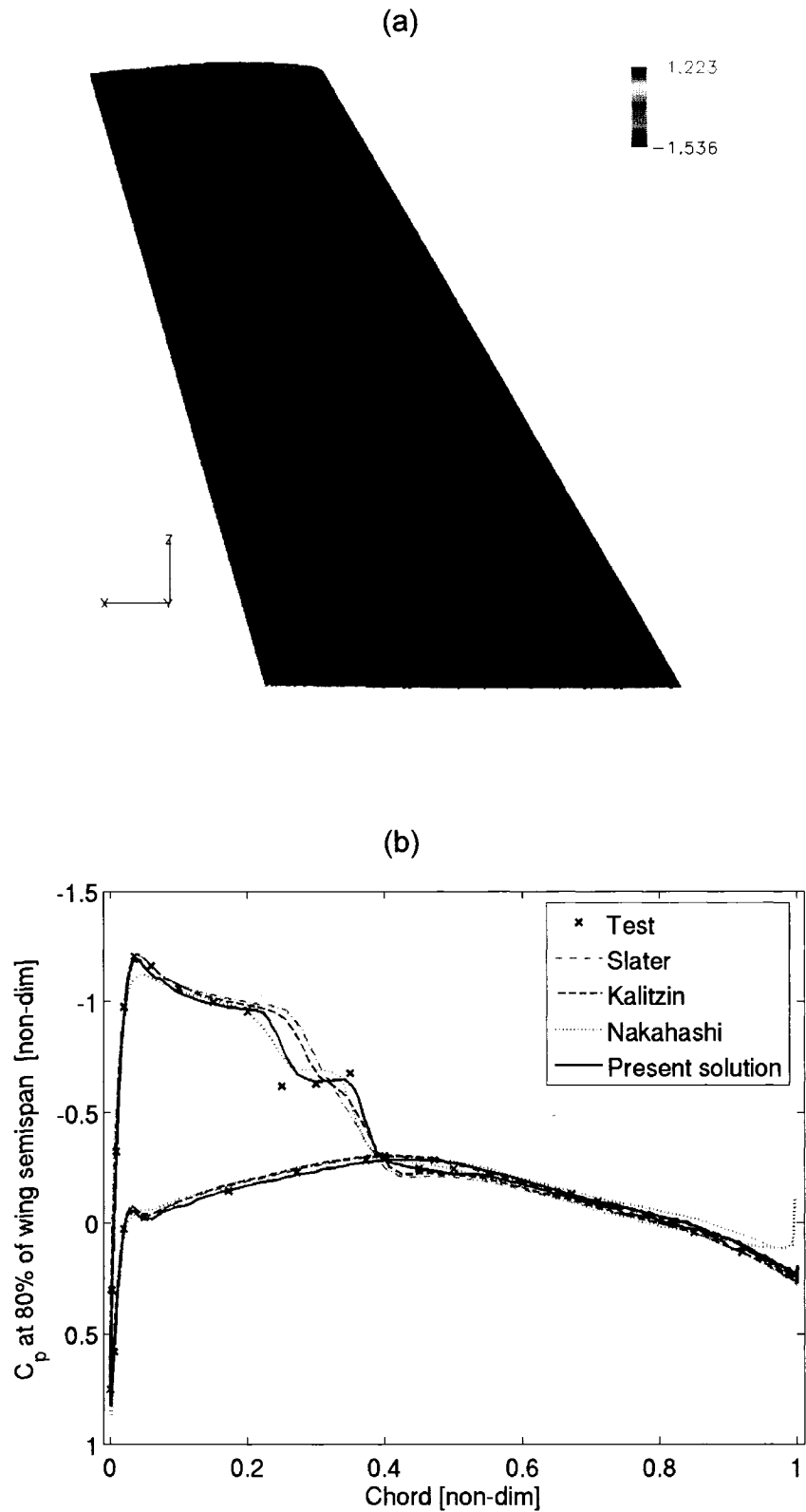


Figure 5-14: 2nd adaptation cycle solution: (a) C_p contours on the wing – top view. (b) Wall C_p comparison at 80% of wing semi span.

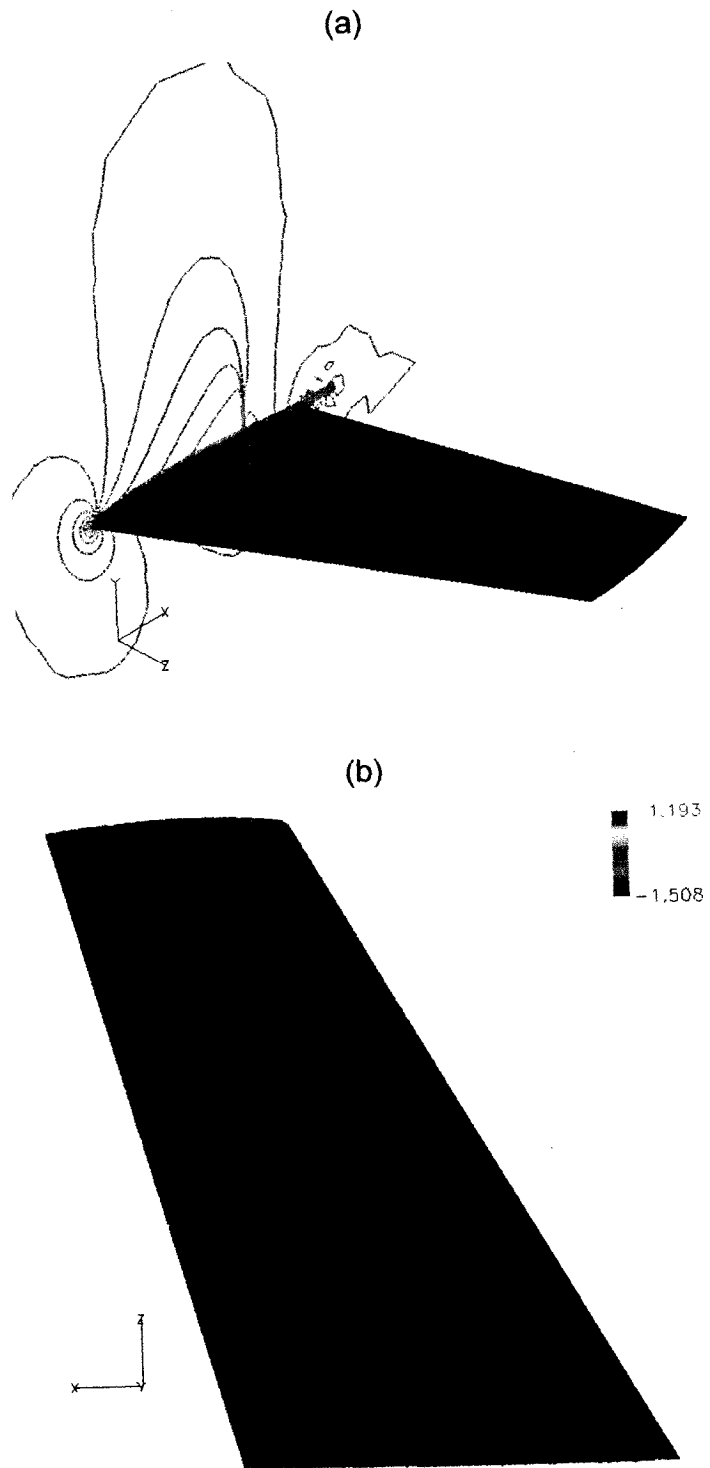


Figure 5-15: 3rd adaptation cycle solution: (a) Isometric view of the wing; C_p distribution on the wing, and Mach contours in the symmetry plane. (b) Top view of the wing with C_p contours.

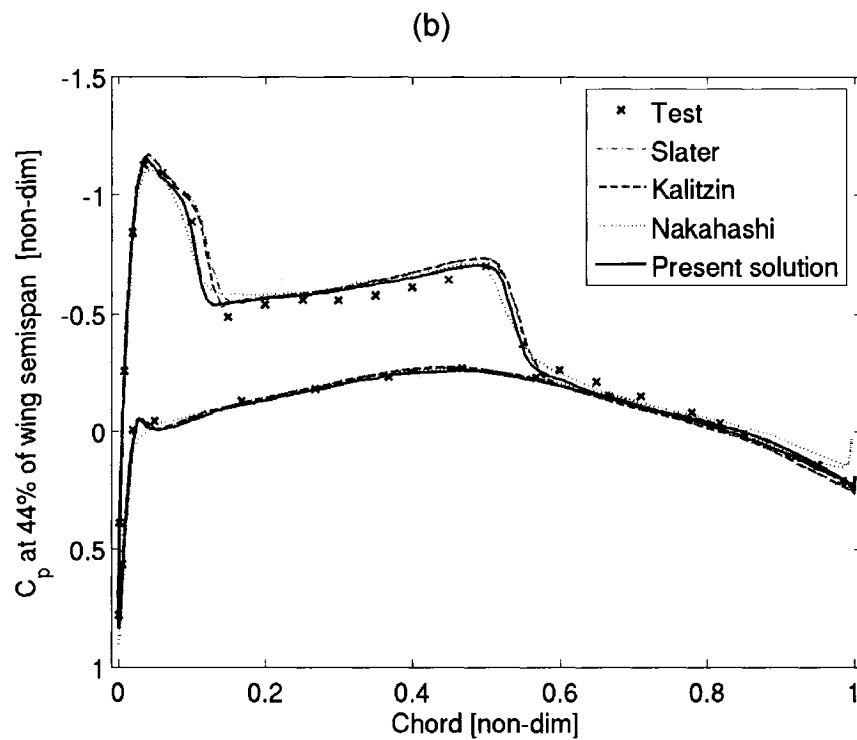
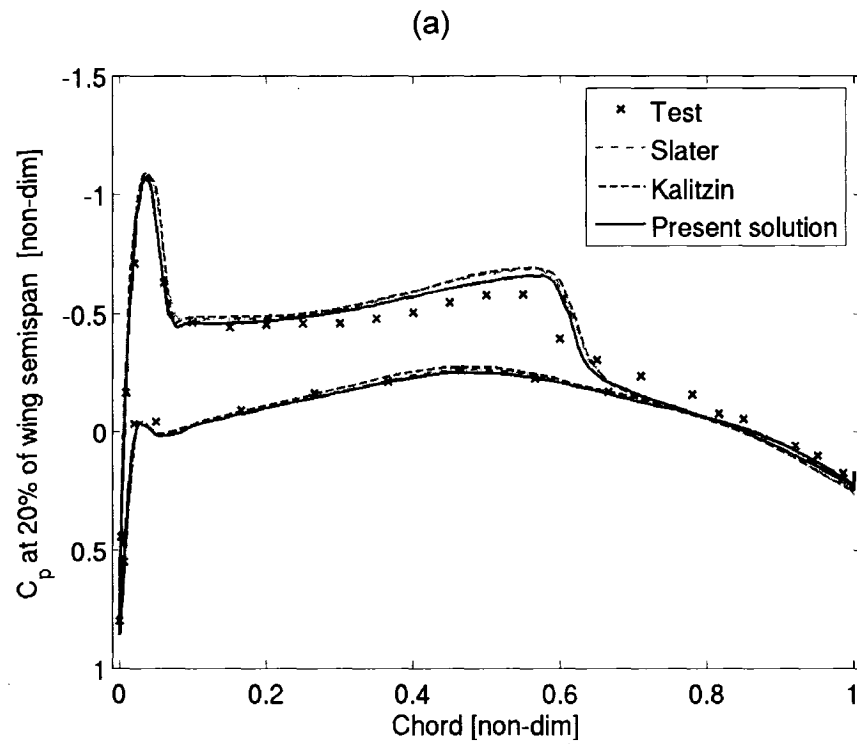


Figure 5-16: 3rd adaptation cycle solution; C_p distribution at the wall – comparison: (a) 20% of the wing semi span. (b) 44% of the wing semi span.

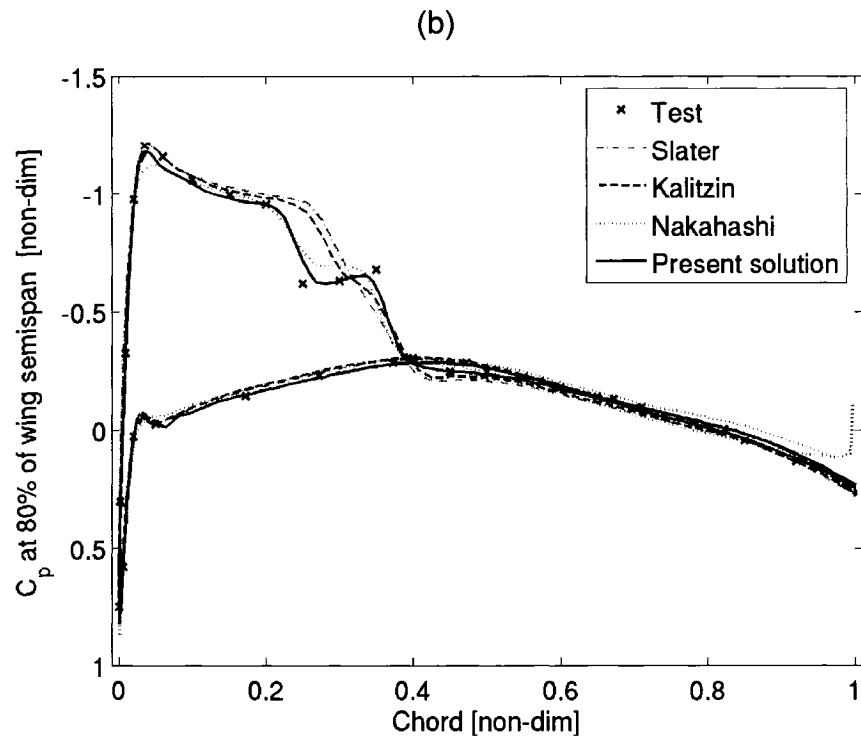
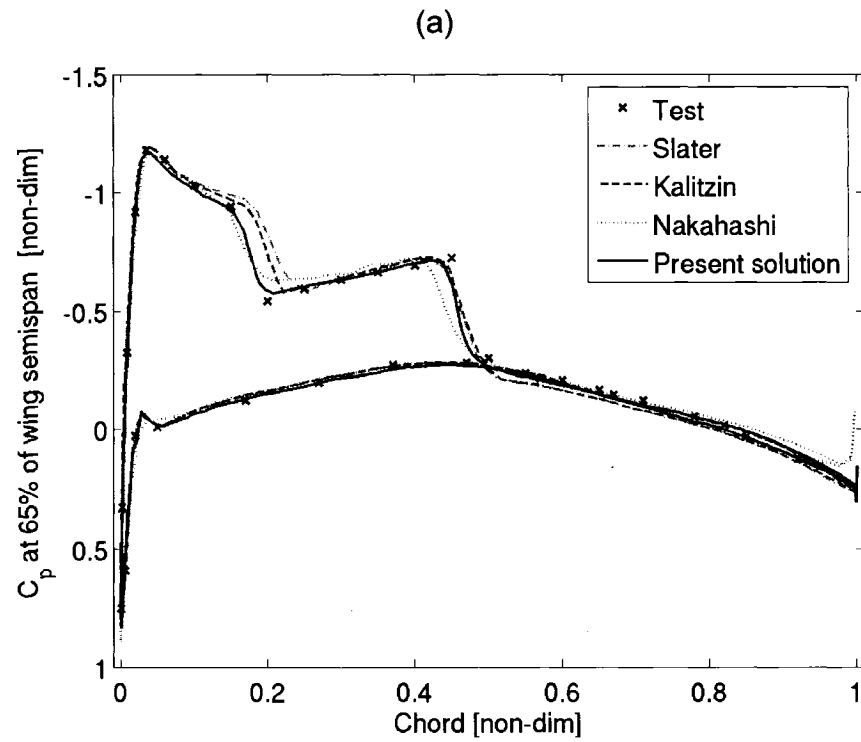


Figure 5-17: 3rd adaptation cycle solution; C_p distribution at the wall – comparison: (a) 65% of the wing semi span. (b) 80% of the wing semi span.

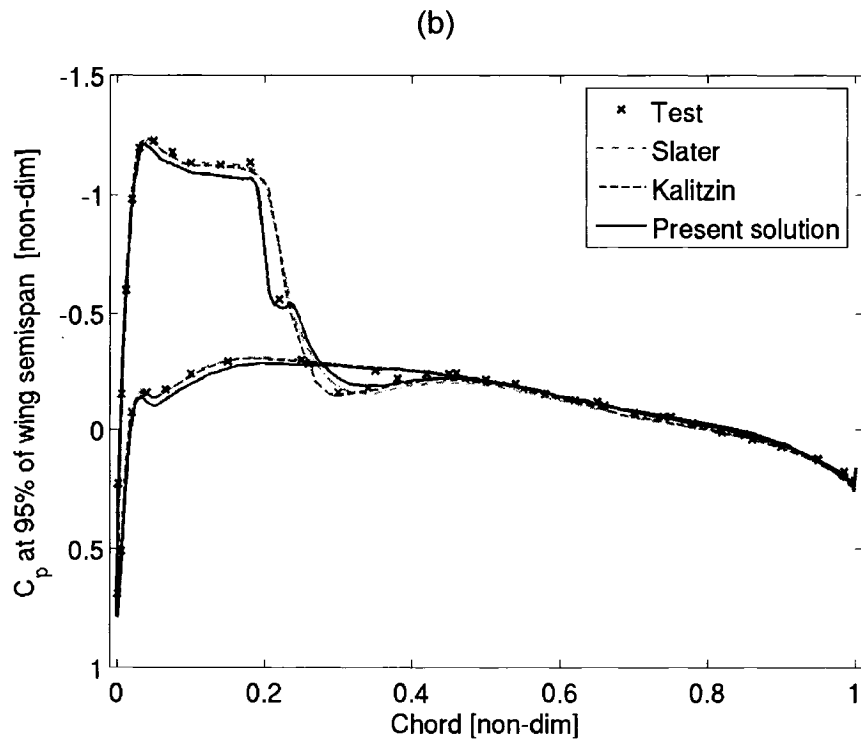
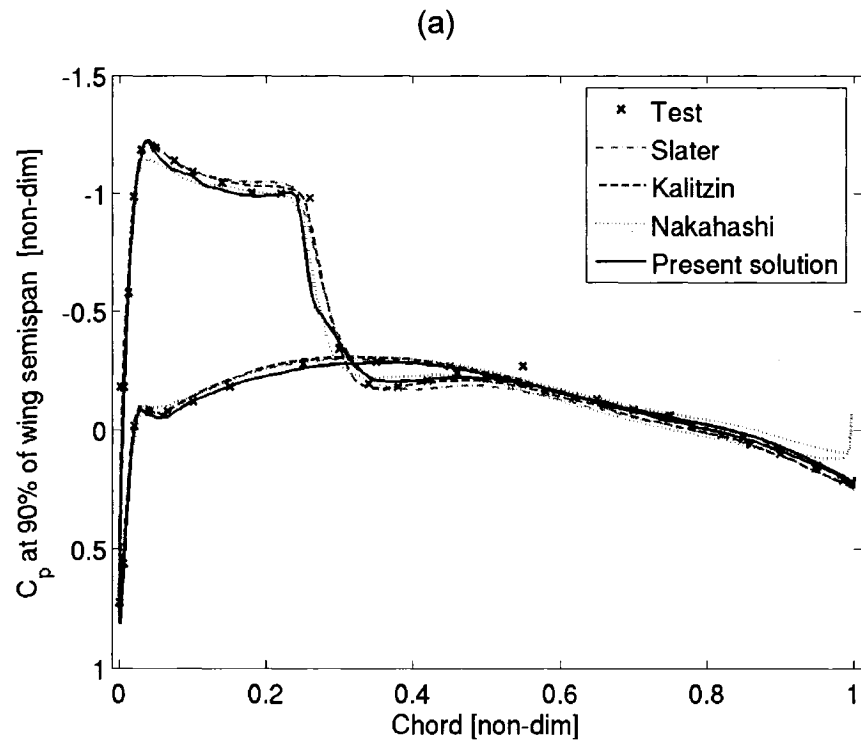


Figure 5-18: 3rd adaptation cycle solution; C_p distribution at the wall – comparison: (a) 90% of the wing semi span. (b) 95% of the wing semi span.

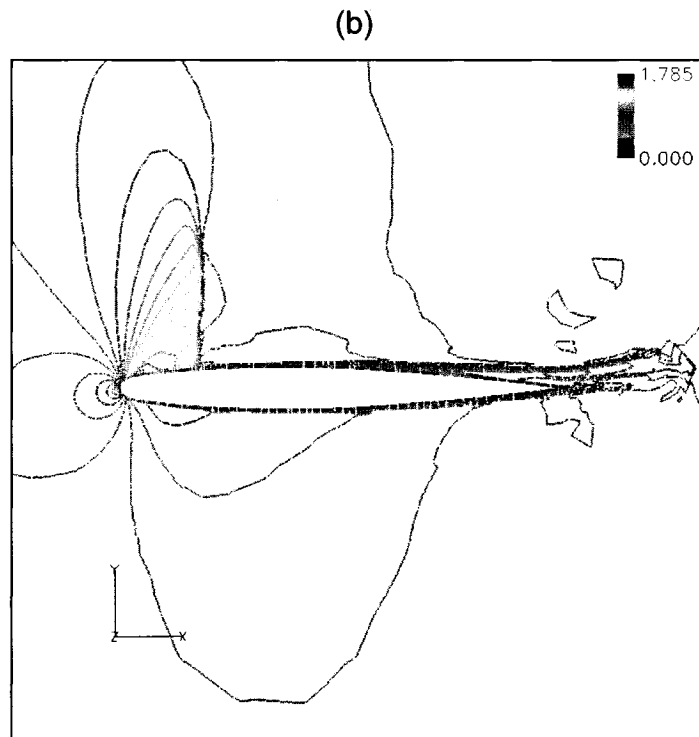
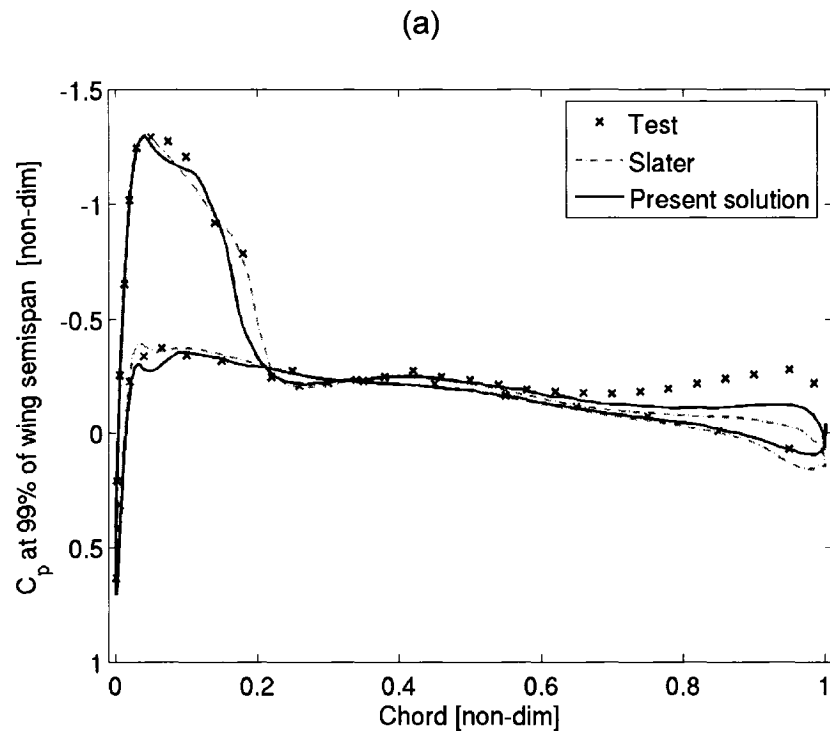


Figure 5-19: 3rd adaptation cycle solution; (a) C_p distribution at the wall, 99% of the wing semi span – comparison. (b) Mach contours at 99% of the wing semi span.

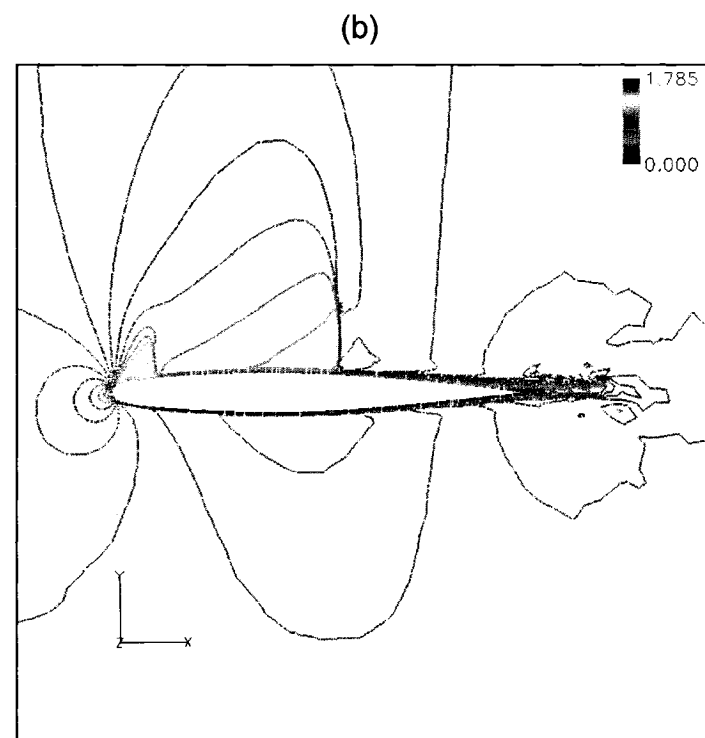
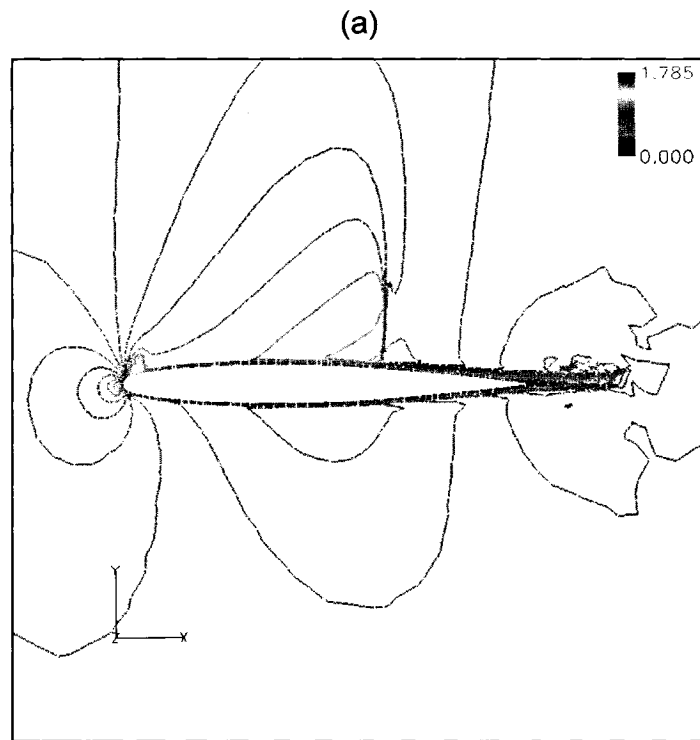


Figure 5-20: 3rd adaptation cycle solution; (a) Mach contours at 20% of the wing semi span. (b) Mach contours at 44% of the wing semi span.

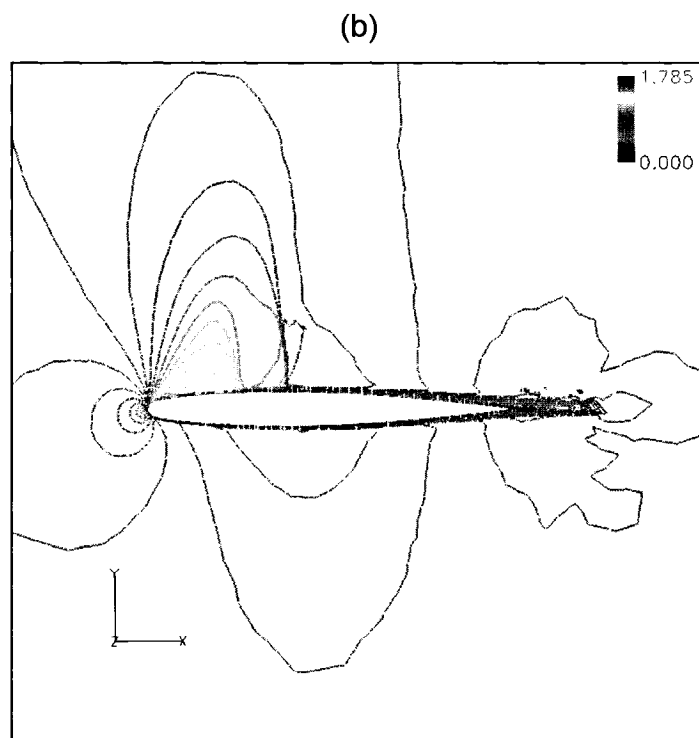
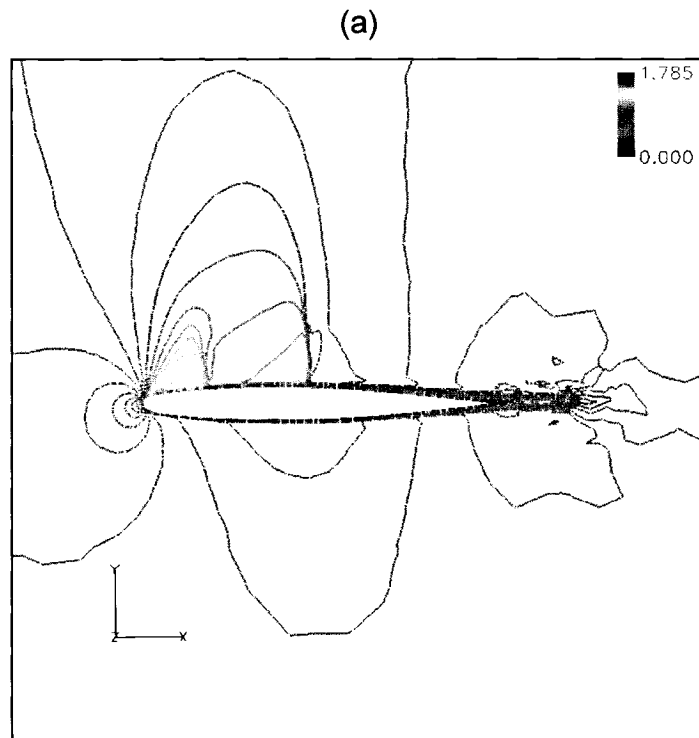


Figure 5-21: 3rd adaptation cycle solution; (a) Mach contours at 65% of the wing semi span. (b) Mach contours at 80% of the wing semi span.

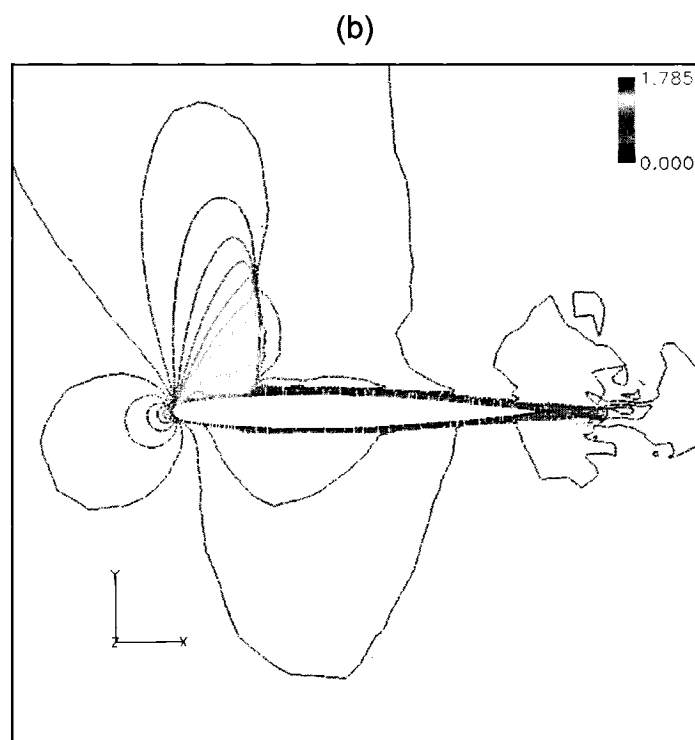
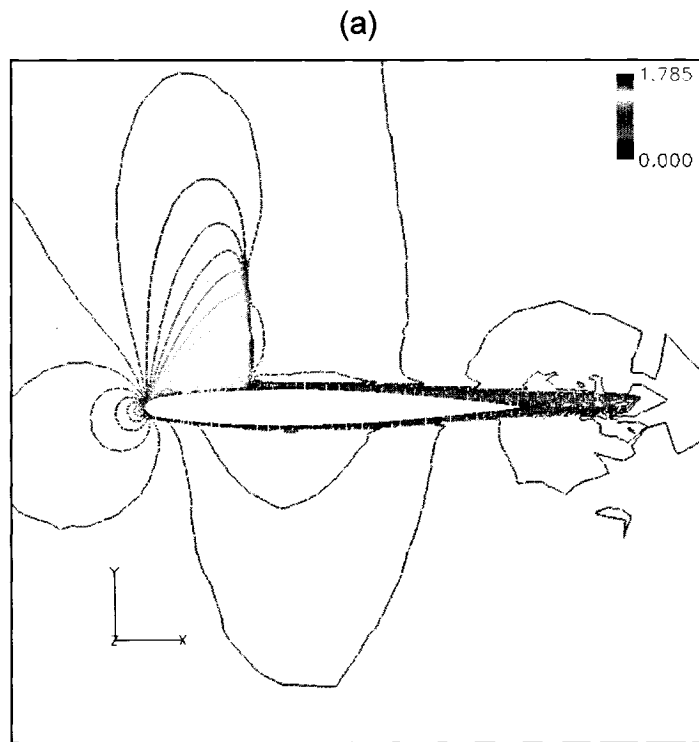


Figure 5-22: 3rd adaptation cycle solution; (a) Mach contours at 90% of the wing semi span. (b) Mach contours at 95% of the wing semi span.

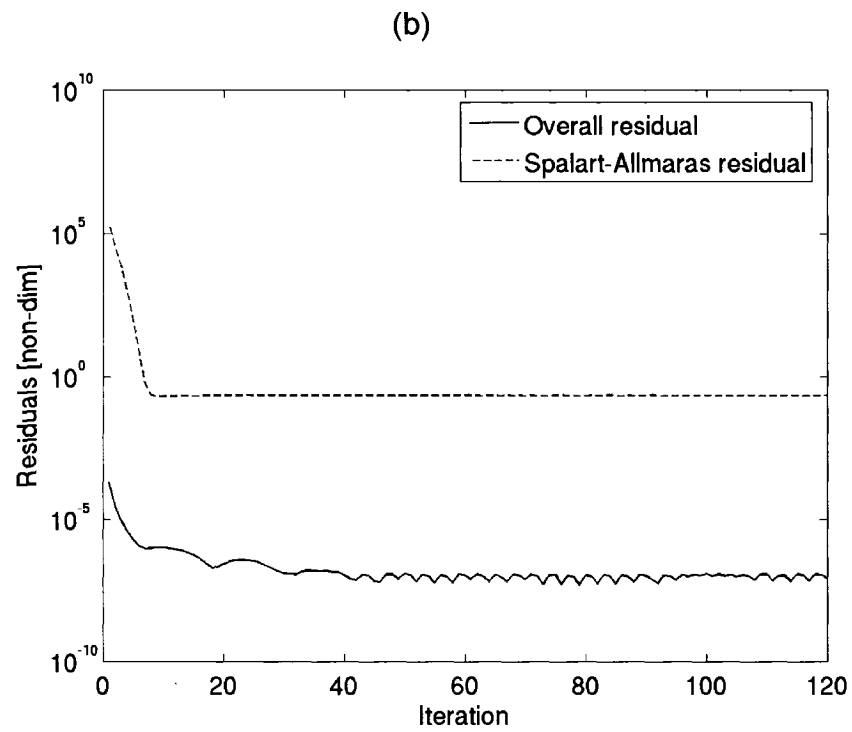
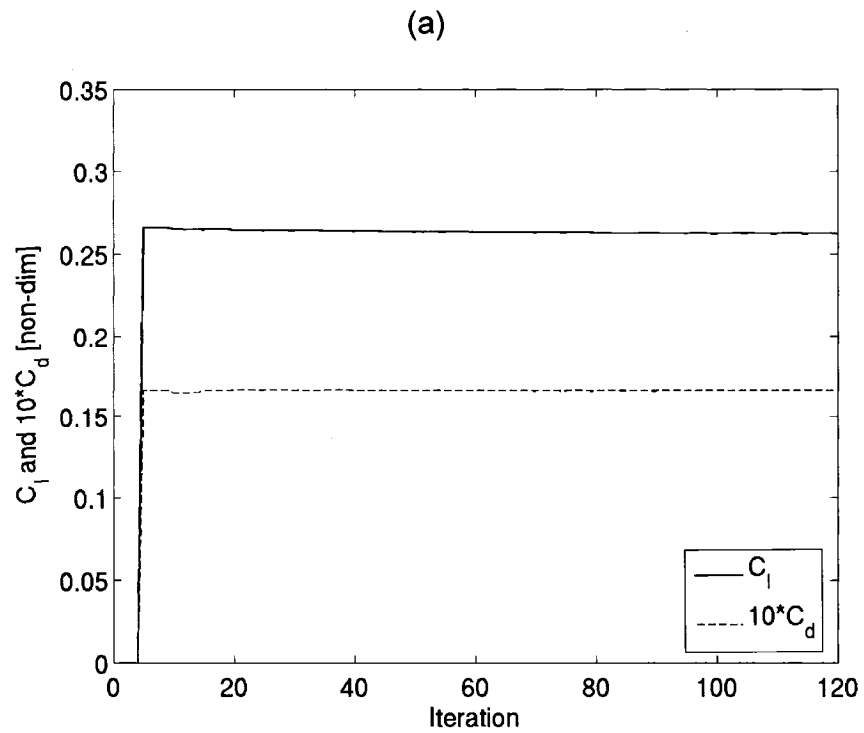


Figure 5-23: 3rd adaptation cycle solution; (a) C_l and C_d histograms. (b) Residuals histograms.

6. Conclusions

In this dissertation we developed an Artificial Viscosity (AV) coupled with a discontinuity detector for the Streamline Upwind Petrov-Galerkin (SUPG) Finite Element Method (FEM) for the compressible Navier-Stokes (N-S) equations. Because AV may modify the actual physics of the fluid flow by generating an over diffused solution, it is important to keep the amount of AV at a minimum while still maintaining a stable scheme. Moreover, in the standard central difference discretization, AV is necessary to avoid odd and even decoupling.

In this work, we propose a systematic way to augment the AV term to produce a sharp shock solution. We introduce an iterative scheme: relatively high AV is used for the first parametric cycle to provide the starting solution that is stable but over-diffused. Then we restart the parametric cycle with less amount of AV. The procedure is repeated until AV is less than certain value. This approach also emphasizes how sensitive to the initial solution the final solution is, when very low AV is employed. Since a good initial guess is required, this procedure has proved to be robust and provided a good approximate solution of the unaltered N-S equations.

There are two types of AV, computed based on local and global derivatives of flow variables. The *first order* AV is computed based purely on local variation, while the *second order* AV employs both local and global variations. It is well known that first order AV leads to sharp shock capturing at the expense of accuracy, while the second order AV leads to accurate solutions but tends to smear the shock. Therefore, in areas of the computational domain that exhibit high gradients of flow properties we select first order AV and obtain first order dominated artificial diffusivity, while in the rest of the domain we select second order AV to get second order dominated artificial diffusivity.

In this dissertation, AV type selection is performed throughout the computational domain and at every simulation time step, employing a discontinuity detector computed based on the normalized second order difference of pressure. The AV amount supplemented into the system is constructed as a blended first and second order AV.

To further improve the quality of the final solution, we also employ mesh adaptation techniques. Mesh points are clustered around flow discontinuities and zones showing high gradients of flow properties, and the mesh becomes more stretched after each adaptation cycle. In the case of a large number of adaptation cycles the cell size becomes small enough to allow for AV be computed just using first order terms, eliminating the need of a shock detector. For a sufficiently refined mesh the AV coefficient can be selected to be zero and the solution is still accurate and stable.

Numerical results of two and three dimensional simulations of transonic viscous flows indicated that shock detector applied to SUPG FEM shows significant improvement for transonic viscous flow simulations and behaves as a limiter to dynamically control AV type. In three dimensional simulations it was necessary to employ mesh adaptation together with shock detection. Mesh adaptation further improved the smoothness of the solution away from the shock, while maintaining sharp discontinuities. Without mesh adaptation in 3D simulations, the solution shows spurious oscillations and the lambda shock wave is not sharp.

This dissertation has shown successful implementation of AV together with mesh adaptation to Streamline Upwind Petrov-Galerkin (SUPG) FEM. While our main focus is limited to fluid flow FEM, the underlined idea in this work is not limited. It can be extended to initial-boundary value problems which contain discontinuities in the domain.

6.1. Future work

The present work on artificial viscosity (AV) selection coupled with mesh adaptation technique for 2D and 3D transonic viscous flows presents good results for RAE 2822 airfoil and ONERA M6 wing test cases. An immediate extension of this work is to solve the transonic viscous flow for a more complex geometry (e.g. DLR transonic wing) and this constitutes work in progress at McGill's CFD Lab. The goal is to obtain a versatile finite element code to be used for industrial transonic aerodynamics problems. However, to improve the

artificial viscosity mechanism discussed in this dissertation, several options may be considered in the future work:

- Jameson [35] indicated four limiters (Minmod, Van Leer, Superbee, alfa-mean) within symmetric limited positive (SLIP) and finite volume (FV) schemes. Based on these limiters, a new shock sensor definition within FEM can be investigated for viscous dominated flows.
- Implementation of a new AV switching logic that insures the second order AV terms are completely turned off in regions with shocks. The outcome expected is a sharper shock and ultimately a better drag and lift prediction.
- Optimization of parameters F , G , S used in AV switching. Moreover, a reduction of the number of parameters used in the present implementation from three to just one will be attractive from an industrial perspective.
- The present solutions were obtained using hybrid meshes (high aspect ratio prismatic elements in the viscous interface and tetrahedrons in the rest of the domain). However, it will be more economic to obtain transonic viscous solutions using tetrahedron grid elements for all computational domain. This approach will require a study of boundary layer resolution and shock capturing mechanisms for this type of mesh.

These recommendations could all reinforce the quality of transonic viscous flow solutions obtained using the present software package.

Bibliography

- [1] Nadarajah, S.K.: *The Discrete Adjoint Approach to Aerodynamic Shape Optimization*, Ph. D. Thesis, Stanford University, January 2003
- [2] Slater, W. J.: *Computational studies performed for the ONERA M6 wing case* NASA John H. Glenn Research Center, August 2002, www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing01/m6wing01.html
- [3] Kalitzin, Georgi: *Application of the v2-f Turbulence Model to Transonic Flows*, Center of Turbulence Research, Stanford University, 2000
- [4] Kazuhiro Nakahashi, Dmitri Sharov, Shintaro Kano and Masatoshi Kodera: *Applications of Unstructured Hybrid Grid Method to High-Reynolds Number Viscous Flows*, Int. J. Numer. Meth. Fluids 31:97-111 (1999)
- [5] Schmitt, V. and Charpin, F.: *Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers*, AGARD Report AR-138, 1979
- [6] Corsini, A.: *A FE Method for the Computational Fluid Dynamics of Turbomachinery*, Lecture note, October 1999
- [7] Tannehill, J. C., Anderson, D. A., Pletcher, R. H.: *Computational Fluid Mechanics and Heat Transfer*, Second edition, Taylor & Francis 1997
- [8] Schlichting, H.: *Boundary-Layer Theory*, 6th edition, McGraw-Hill, New York, 1968
- [9] Newmerical Technologies Int.: *FENSAP – user's guide*, versions 2003R1 2005R1 Beta2, Montreal, March 2003 & 2005
- [10] Rumsey, C.L. and Vatsa V.N.: *A comparison of the predictive capabilities of several turbulence models using upwind and central difference computer codes*, AIAA-93-0192, 1993
- [11] Hughes, T.J.R.: *The Finite Element Method – Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, INC., New Jersey, 1987
- [12] Hirsch, C.: *Numerical Computation of Internal and External Flows*, vol. 1, John Willey & Sons, 2001

- [13] Brooks, A.N., Hughes, T.J.R.: *Streamline Upwind/Petrov-Galerkin Formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering 32(1982) 199-259, 1982
- [14] Jameson, A., Schmidt, W., Turkel, E.: *Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes*, AIAA 1981-1259, 1981
- [15] Carafoli, E., Constantinescu, V.N.: *Dinamica Fluidelor Compresibile*, pg. 30-33, 141-149, Editura Academiei Republicii Socialiste Romania, Bucuresti 1984
- [16] Cook, P.H., McDonald, M.A., Firmin, M.C.P.: *Aerofoil RAE 2822 – Pressure Distributions Boundary Layer and Wake Measurements*, Experimental Data Base for Computer Program Assessment, AGARD Report AR 138, 1979
- [17] Remaki, L., Nadarajah, S., Habashi, W.G., Bogstad, M.C., Kho, C., Mokhtarian, F.: *Mesh Adaptation Impact on Lift and Drag Coefficients*, NSERC-J. Armand Bombardier Industrial Research Chair of Multidisciplinary CFD at McGill, 2004
- [18] Jobe, C.E.: *Thrust and Drag: Its Prediction and Verification*, Progress in Astronautics and Aeronautics Series, vol. 98, AIAA, Washington, 1985
- [19] Christie, I., Griffiths, D.F., Mitchell, A.R., Zienkiewicz, O.C.: *Finite element methods for second order differential equations with significant first derivatives*, Internat. J. Numer. Methods Engrg. Vol. 10, pp1389-1396, 1976
- [20] Heinrich, J.C., Huyakorn, P.S., Zienkiewicz, O.C., Mitchell, A.R.: *An upwind finite element scheme for two dimensional convective transport equation*, Internat. J. Numer. Methods Engrg. Vol. 11, pp134-143, 1977
- [21] Hughes, T.J.R.: *A simple scheme for developing upwind finite elements*, Internat. J. Numer. Methods Engrg. Vol. 12, pp1359-1365, 1978
- [22] Hughes, T.J.R., Atkinson, J.: *A variational basis for upwind finite elements*, IUTAM Symposium on Variational Methods in the Mechanics of solids, Northwestern University, Evanston, IL, 1978

- [23] Johnson, C.: *Finite element methods for convection-diffusion problems*, Fifth International Symposium on Computing Methods in Engineering and Applied Sciences, INRIA, Versailles, 1981
- [24] Nävert, U.: *A finite element method for convection-diffusion problems*, PhD Thesis, Dept. of Computer Sciences, Chalmers Univ. of Technology, Göteborg, Sweden, 1982
- [25] Hughes, T.J.R., Tezduyar, T.E.: *Finite Element Methods for first order hyperbolic systems with particular emphasis on the compressible Euler equations*, Computer Methods in Applied Mechanics and Engineering 45 pp 217-284, 1984
- [26] Babuška, I.: *The finite element method with Lagrangian multipliers*, Numerische Mathematik, 20, pp179-192, 1973
- [27] Brezzi, F.: *On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers*, RAIRO Analyse Numérique, R2 pp129-151, 1974
- [28] N'dri, D., Garon, A., Fortin, A.: *A new stable space-time formulation for two-dimensional and three-dimensional incompressible viscous flow*, Internat. J. Numer. Methods Fluids. Vol. 37, pp865-884, 2001
- [29] Hughes, T.J.R., Franca, L.P., Balestra, M.: *A new Finite Element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations*, Computer Methods in Applied Mechanics and Engineering 59 pp 85-99, 1986
- [30] Hughes, T.J.R., Mallet, M., Mizukami, A.: *A new finite element formulation for computational fluid dynamics: II. Beyond SUPG*, Computer Methods in Applied Mechanics and Engineering 54 pp 341-355, 1986
- [31] Hughes, T.J.R., Mallet, M.: *A new finite element formulation for computational fluid dynamics: IV. A discontinuity capturing operator for multidimensional advective-diffusive systems*, Computer Methods in Applied Mechanics and Engineering 58 pp 329-336, 1986

- [32] Almeida, R.C., Galeão, A.C.: An adaptive Petrov-Galerkin formulation for the compressible Euler and Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering* 129 pp 157-176, 1996
- [33] John D. Anderson, Jr.: *Fundamentals of Aerodynamics*, second edition, McGraw-Hill, Inc., 1991
- [34] Martinelli, L.: *Calculations of Viscous Flow with a Multigrid Method*, PhD thesis, Princeton University, Princeton, NJ, October 1987.
- [35] Jameson, A.: *Analysis and design of numerical schemes for gas dynamics 1 artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence*, *Int. J. of Comp. Fl. Dyn.*, vol. 4, pp. 171-218, 1995.

7. Appendix A – RAE 2822 measured coordinates, [16]

x/c	z/c lower	z/c upper
0.00000	0.00000	0.00000
0.00060	0.00317	0.00323
0.00241	0.00658	0.00642
0.00541	0.00957	0.00945
0.00961	0.01273	0.01269
0.01498	0.01580	0.01579
0.02153	0.01880	0.01875
0.02923	0.02180	0.02163
0.03806	0.02472	0.02445
0.04801	0.02761	0.02726
0.05904	0.03042	0.03004
0.07114	0.03315	0.03280
0.08427	0.03584	0.03552
0.09840	0.03844	0.03817
0.11349	0.04094	0.04073
0.12952	0.04333	0.04321
0.14645	0.04561	0.04558
0.16422	0.04775	0.04778
0.18280	0.04977	0.04987
0.20215	0.05167	0.05187
0.22221	0.05340	0.05377
0.24295	0.05498	0.05556
0.26430	0.05638	0.05713
0.28622	0.05753	0.05848
0.30866	0.05843	0.05967
0.33156	0.05900	0.06070
0.35486	0.05919	0.06155
0.37851	0.05893	0.06220
0.40245	0.05817	0.06263
0.42663	0.05689	0.06285
0.45099	0.05515	0.06286
0.47547	0.05297	0.06261
0.50000	0.05044	0.06212
0.52453	0.04761	0.06135
0.54901	0.04452	0.06030
0.57336	0.04127	0.05895
0.59754	0.03791	0.05733
0.62149	0.03463	0.05547
0.64514	0.03110	0.05339

x/c	z/c lower	z/c upper
0.66845	0.02770	0.05112
0.69134	0.02438	0.04857
0.71378	0.02118	0.04612
0.73570	0.01812	0.04338
0.75705	0.01524	0.04075
0.77778	0.01256	0.03795
0.79785	0.01013	0.03514
0.81720	0.00792	0.03231
0.83578	0.00594	0.02948
0.85355	0.00422	0.02670
0.87048	0.00273	0.02397
0.88651	0.00149	0.02131
0.90160	0.00049	0.01874
0.91574	-0.00027	0.01627
0.92886	-0.00081	0.01393
0.94096	-0.00113	0.01170
0.95200	-0.00125	0.00964
0.96194	-0.00125	0.00775
0.97077	-0.00113	0.00606
0.97847	-0.00094	0.00455
0.98502	-0.00071	0.00326
0.99039	-0.00048	0.00218
0.99459	-0.00026	0.00132
0.99759	-0.00009	0.00069
0.99940	0.00001	0.00030
1.00000	0.00000	0.00000

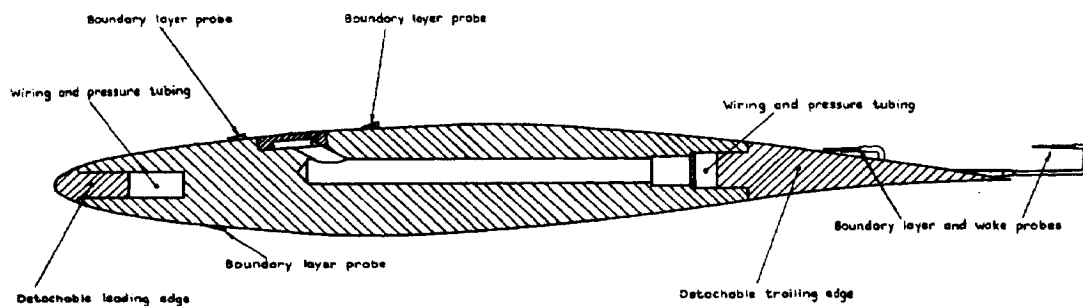


Figure 7-1: RAE 2822 transonic airfoil – model cross section, [16]

8. Appendix B – ONERA M6 wing data, [5]

x/l	z/l		x/l	z/l
0.0	0.0		0.3761446	0.0489296
0.0000165	0.0006914		0.4019567	0.0499272
0.0000696	0.0014416		0.4274223	0.0484833
0.0001675	0.0022554		0.4528441	0.0479351
0.0003232	0.0031382		0.4781197	0.0471661
0.0005509	0.0040959		0.5032514	0.0461903
0.0008657	0.0051343		0.5282426	0.0450202
0.0012868	0.0062598		0.5530937	0.0436741
0.0018364	0.0074784		0.5778043	0.0421584
0.0025441	0.0087958		0.6023757	0.0405241
0.0034428	0.0102163		0.6268104	0.0387613
0.0045704	0.0117419		0.6511093	0.0368990
0.0059751	0.0133708		0.6752726	0.0349542
0.0077112	0.0150951		0.6993027	0.0329402
0.0098413	0.0168984		0.7231995	0.0308662
0.0124479	0.0187537		0.7469658	0.0287365
0.0156171	0.0206220		0.7705998	0.0265505
0.0194609	0.0224545		0.7941055	0.0243027
0.0241067	0.0242004		0.8174828	0.0219942
0.0297008	0.0258245		0.8407324	0.0195838
0.0364261	0.0273317		0.8638564	0.0170915
0.0444852	0.0287912		0.8868235	0.0145051
0.0541248	0.0303278		0.9061905	0.0122389
0.0656303	0.0320138		0.9225336	0.0102727
0.0793366	0.0338372		0.9363346	0.0085827
0.0956354	0.0357742		0.9479946	0.0071423
0.1149796	0.0377923		0.9578511	0.0059224
0.1378963	0.0398522		0.9661860	0.0048907
0.1649976	0.0419080		0.9732361	0.0040190
0.1919327	0.0436214		0.9792020	0.0032796
0.2187096	0.0450507		0.9842508	0.0026547
0.2453310	0.0462358		0.9885252	0.0021257
0.2717978	0.0471987		0.9921438	0.0016778
0.2981113	0.0479494		0.9952080	0.0012985
0.3242726	0.0484902		0.9979030	0.0009773
0.3502830	0.0488183		1.0000000	0.0007052

Figure 8-1: Design values for root section coordinates of the symmetrical profile

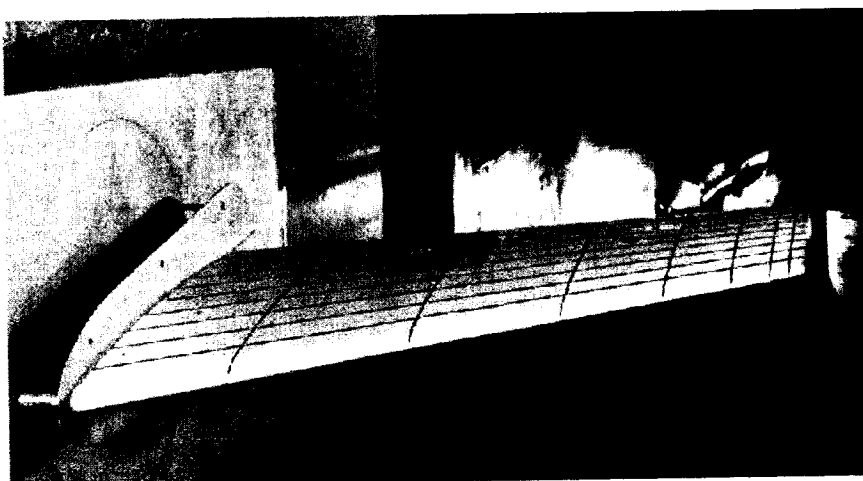


Figure 8-2: ONERA M6 semi-span wing

SWEPT WING M6

Aspect ratio $A = 3.8$
 Taper ratio $\lambda = 0.56$
 Sweep angle $\Lambda_{25\%} = 26.7^\circ$

ROWS OF PRESSURE TAPS

N°	y/b	upper	under
1	0.20	23	11
2	0.44	23	11
3	0.65	23	11
4	0.80	23	11
5	0.90	31	14
6	0.95	31	14
7	0.99	31	14

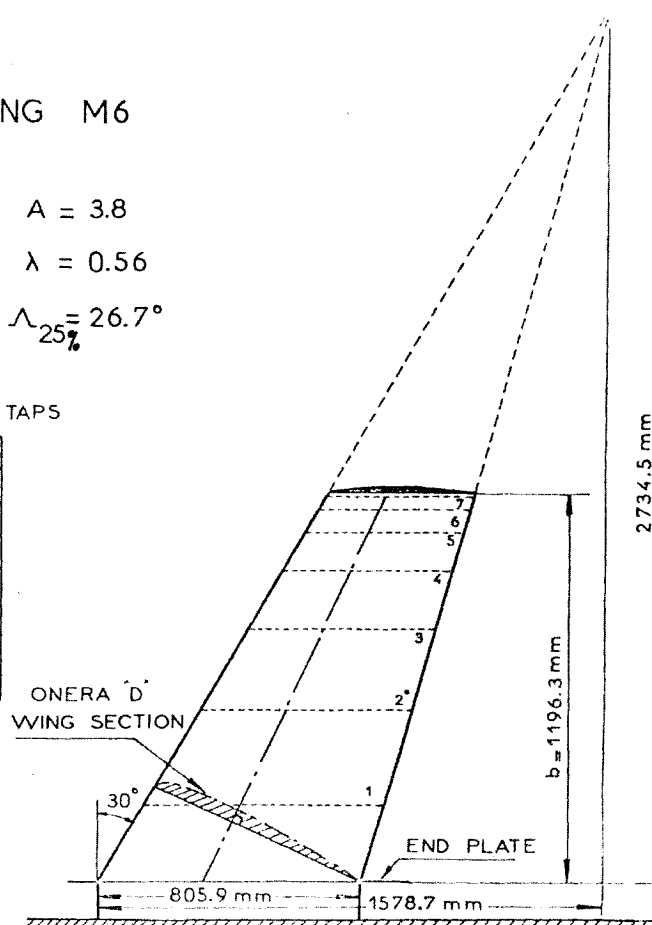


Figure 8-3: Additional remarks for ONERA M6 wing